University of Chittagong

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Project on

Developing an e-blood system using object-oriented software engineering approach

Submitted To:

A.H.M Sejedul Haque

Assistant Professor

Dept. Computer Science &Engineering

University of Chittagong

Submitted By:

Tamima Nayam

ID-11205047

CSE 4th Year

Session: 2010-2011

Dept. Computer Science &Engineering

University of Chittagong

Date:23/10/2016

DEDICATION

To my dearest parents

To all of my honorable teachers

To all of my friends and my beloved

Declaration

I declare that except where otherwise stated, this project is entirely my own work and has not been submitted in any form to anywhere for any degree.

……

Tamima Nayam

(Candidate)

APPROVAL

The project titled “Developing an e-blood system using object-oriented software engineering approach” submitted by Tamima Nayam,, ID No: 11205047 session: 2010-2011 has been accepted as satisfactory for the fulfillment of the requirements for the degree of B.Sc. Engineering in Computer Science and Engineering on University of Chittagong.

…….

Signature of Supervisor

A.H.M Sajedul Hoque

Assistant Professor

Computer Science and Engineering

University of Chittagong

ACKNOWLEDGEMENTS

In the very beginning, I wish to express my deep sense of gratitude to the Almighty Allah, for giving me the strength to fulfill this project successfully. I wish to express my deepest gratitude and indebtedness to my supervisor **A.H.M. Sajedul Hoque,** Assistant Professor**,** Department of Computer Science and Engineering, for suggesting the problem, constant supervision, encouragement, criticism and kind assistance at all stage of this project work.

I would like to thank to our honorable Chairman, the Department of Computer Science and Engineering, for providing me all possible laboratory facilities to do this piece of research work.

I express my sincere gratitude to our honorable teachers for suggestions and various sorts of help for this project.

I also like to express my thanks to all CSE 11th batch students who have shared their knowledge with me and has given their inspiration and encouragement during the project work.

Since this is the first attempt to automate any manual system, something may be missing. Finally I would like to declare that the responsibilities of all missing that still remain would be correct in future.

Date: 19/03/17 Tamima Nayam

ABSTRACT

An e-blood system is a cache or bank of blood or blood components, gathered as a result of blood donation or collection, stored and preserved for later use in blood transfusion. The purpose of this study was to develop a blood management information system to assist in the management of blood donor records and ease/or control the distribution of blood in various parts of the country basing on the hospital demands. Without quick and timely access to donor records, creating market strategies for blood donation, lobbying and sensitization of blood donors becomes very difficult. The blood management information system offers functionalities to quick access to donor records collected from various parts of the country. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked. It provides to management timely, confidential and secure medical reports that facilitates planning and decision making and hence improved medical service delivery. The reports generated by the system give answers to most of the challenges management faces as far as blood donor records are concerned. But in manual system is too much analogues. If any person requests a query about a product, then the responsible employee can't give him answer in a quick way. Watching those problems we have tried to develop new software named e-blood system. For developing this software we use the **PHP** on the front end to develop the user interface of the e-blood software and the database is designed in MySQL as the backend.We tried our best to include all the components, which actually needed in a Blood Bank.

Tables of Contents

Chapter 1: Introduction
1.1. Intoroduction……8
1.2. Motivation……8
1.3. Methodology……...10
1.4. Structure of the project……11
Chapter 2: Software Requirement Specification(SRS)
2.2.User requirement of the system……..12
2.2.1.Accepting and storing information……..12
2.2.2.Dispaly donor's list……..13
2.2.3.Display receiver's list……...…13
2.2.5.Contact donor……13
2.3.System requirement specification……13
2.4.System Scenario's……13
2.5.Use cases……..15
2.5.1.Use cases for donor and receiver of the system……....15
2.5.2.Use cases for admin of the system……...….21
2.6.Use case diagram……..25
2.6.1.Use case for donor and receiver modules…….26
2.6.2.Use case for admin module……...…27
2.7.DFD of the system……...…….27
2.8.Conclusion……...…….29
Chapter 3: Detailed Design Document(DDD)
3.1.Activity Diagram……..……...30
3.1.1.Activity Diagram for Donor and Receiver/Patient….....30
3.1.2.Activity Diagram for Admin.……...…….37
3.2. Conceptual Class Diagram……..……43
3.3.Sequence Diagram……43
3.3.1.Sequence Diagram for Donor and Receiver……..43
3.3.2Sequence Diagram for admin panel……53
3.4.Architectural Pattern…….60
3.5.Entity-Relationship Diagram(ERD)…….62
3.6.Conclusion…….63
Chapter 4: Implementation
4.2.Detailed Class Diagram……..……..…...64
4.2.1Detailed Class Diagram For Donor and Receiver…….….....65
4.2.2.Detailed Class Diagram for Admin……...66
4.3.Used Design Pattern……..……...67
4.4.Used Software Process model……..67
4.5.Demonstration of the project……68
4.6.Testing……..74
4.6.Conclusion……78
Chapter 5:Conclusion
5.1. Result and discussions……79
5.2. Limitation of the project…….80
5.3. Future work…….80
5.4. Conclusion……..80

CHAPTER 1: INTRODUCTION

1.1. Introduction

Blood is a saver of all existing lives. So in such emergency cases, it is difficult for hospital staff to collect blood in case of shortage of blood without having appropriate resources. In this project an e-blood system has been developed using object-oriented software engineering approach which provides an online platform for a quick access to the required donor[1]**.** Using this system**,** users can view the information about registered blood donors such as name, address, and other such personal information along with their details of blood group and other medical information of donor.

1.2. Motivation :

Every second of every day ,people around the world of all ages and from all walks of life need a blood transfusion to survive. Currently, the public can only know about the blood donation events through conventional media means such as radio, news paper or television advertisements. Even if there is electronic means, it is only used to publicize about that hospital or medical center blood donation drives provided if that hospital or medical center is having an online portal. There is no nation wide information regarding the blood donation drive available on any of the portal. Besides, for those who want to make blood donation, they cannot make early reservation or booking on the session and day that they are free online. It is a very important facility for those who are very busy and yet enthusiastic people to know and be sure when they can make blood donation rather than trying to figure out where and when they can make blood donation when they are free.

A. Problem Statement:

In the current days, it was such that everything was done manually i.e. maintaining records, maintaining accounts and particulars. It is a burden and huge headache for the people to handle each and every minute detail. Moreover it is not possible for any human being to update thousands of information ever.

Without an automated management system, there are also problems in keeping track of the actual amount of each and every blood type in the blood bank .Man-made error such as forget to record the usage or input of the blood can cause the inaccuracy in the amount of certain blood type available in the blood bank. In addition, there is also no alert available when the blood quantity is below its par level.

B. Objectives:

After defining the problem statement in the current system, The main objective of the study was to create electronic blood donor management information system in order to assist in the management of blood donor records, screening of blood, planning and share information in a more confidential, convenient and secure way using modern technology. This automated system ensures appropriate computerized mechanism with proper security to store data. This information would be used in identifying/locating existing BDs (Blood Donor), carrying out pre-donation counseling and taking blood results. Unauthorized persons however, easily accessed the paper system and hence making it impossible to keep secrecy and confidentiality expected of medical records. The security of the medical records was also not inadequate as any person could easily access them. I want to develop such a system, which will satisfy some aspects such as reliability, maintainability, cost-effectiveness, security and user-friendliness.

C.Preliminary Solution:

As the manual system has so many limitations, one way to solve this problem is to digitalize the process. A web based application can be developed to overcome the limitations. In this regard, we will need **a database server**, **a good internet connection** and **minimum 5 workstation.**

D.Project Scope:

The deliverable of this project is the “E-BLOOD” system, which is a web-based database application system. The scope of the project will cover the system functionalities, technologies used, the targeted users, system deployment and methodology.

In this project the methodology to be used in developing “e-BLOOD” system, namely the **Object Oriented Analysis and Design (OOAD)** . Instead of **structured System analysis and design(SSAD)** we feel better and approach by Object Oriented Analysis and Design(OOAD).Because, OOAD significantly simplifies the development of the system compare to SSAD .The development time, the level of organization ,the robustness, and the code reuse are all greatly enhanced by the OOAD methodology. In OOAD ,there is no separation between the analysis and design phases, which improves communication between the users from beginning to end of project development. This methodology also reduce design ,programming and validation costs..

1.3. Methodology

Methodology is needed for software in order to build it with consistency. At first I have to decide the basic theory and then realize its necessity. Analyzing about points, I am completely aware of my purpose, my required and proposed system of Blood Bank Management. To determine by how I will proceed successfully, I will introduce a methodology, which contains the steps for developing my software depending on proposed system. Methods are simply techniques that I apply to accomplish some objectives. Methods include guidelines, techniques and tools for building a software system. In order to develop an automated e-Blood system, I need a methodology. A **Software Development Life Cycle** (SDLC) is an abstract description of the structured and methodological development and modification process applied to main stage of producing and developing software. In this project I would like to approach by waterfall model which is appropriate framework at this point. Because, waterfall is a linear approach to software development.It is simple to implement and also the amount of resources required for it are minimal.In this model,output is generated after each stage(as seen before),therefore it has high visibility.Project management,both at interval level and client’s level is easy again because of visible outputs after each phase[2] [3]..Here, deadline can be set for the completion of each phase and evaluation can be done from time to time,to check if project is going as per milestones.This methodology is significantly better than the haphazard approach to develop software.It provides

a template into which methods of analysis,design,coding.testing and maintenance can be placed.Moreover,this methodology is preferred in projects where quality is more important as compared to schedule or cost.

1.4. Structure of the Report

Effective management of a software project depends on thorough planning the progress of the project. The project developer must anticipate the problems that might arise and prepare tentative solutions to those problems. A plan, draw up at the start of the project, should be used as the driver for the project. This initial plan should be the best possible plan given the available information. It involves as the project progresses and better information becomes available. This project is divided into several segments. They are:Feasibility study,Software Requirements Specification, Design, Implementation, Testing and Maintenance . In this project,through the feasibility study the main objectives of this chapter is to understand the necessity of automation, it will provide to know the whole overview of the present and proposed e-blood System. and also be aware of drawbacks of the manual system. Consequently this documentation will provide a very important chapter which named Software Requirements Specification .The pupose of this chapter is to describe a number of generic requirements engineering processes for e-blood system. It will also allow to understand the principal requirements engineering activities and their relationship which have been introduced to several techniques of requirements analysis and also to be understand importance of requirements validation and how requirements validation and how reviews are used in this process. Then the design chapter will describe the techniques which are revolted according to requirement analysis, specification and validation. This part outcomes a detailed focus on software development. Most important chapter of this project is implementation it contains coding, testing, object models of the software. The last chapter contains application and maintenance of the system that implies effectiveness and reliability of the software.

CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION

2.1. Introduction:

This chapter aims at defining overall software requirement for e-blood system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate. This document is intended for both the user and the developers of the system.

This system is designed to allow to communicate among a group of donors and receivers so that they can get blood in emergency. This system will contain a database server containing all the information of donors. Here all the information will remain safe and under maximum security. This system will facilitate communication between donors and receivers very easily.It will make blood donation a lot easier and it will save millions of lives all around the world.

In this chapter the content that have been discussed in this report are state below:

- Requirements of the system.
- Scenario of the system.
- Use cases of the system.
- Use case Diagram.

2.2. User Requirements of the system:

User has some requirements of this system which are pointed below:

1. Accepting and storing information.
2. Display donor's list.
3. Display receiver's list
4. Concurrent access.
5. Contact donor.

2.2.1.Accepting and storing information:

- System accepts Donor's information including name, contact, e-mail, date of birth, blood group, present address and stores it.

- System accepts Receiver's information including name, contact, required blood group, quantity of blood, address location amd stores it.

2.2.2.Display Donor's list: System will show the donor's list to the user.

2.2.3.Display Receiver's list: System will show the receiver's list to the user.

2.2.4.Concurrent access: Many users can use system concurrently without noticeable system

2.2.5.Contact Donor: User can contact donor either by message or phone call.

2.3. System Requirement Specification:

The requirement for the system descriptions of the services provided by the system and operational constraints. User requirements are statements, of what services the system if expected to provide and the constrains under which it must operate. User requirement of 'e-Blood' system(EBS) is given below:

- EBS should provide the facility of the user to act as a donor in the system.
- EBS should provide the facility of the user to see donor information in the system.
- EBS should provide the facility of the user to act as a receiver in the system.
- EBS should provide the facility of the user to see receiver information in the system
- EBS should provide the facility of the user to update information of the donor.
- EBS should provide the facility of the user to update information of the receiver.
- EBS should provide the facility of the user to contact donor via phone or messages.

2.4. System's Scenario's:

Scenarios can be particularly useful for adding details to an outline requirements description. People usually finds it easier to relate real life examples rather than abstract descriptions. They can understand and criticize a scenario of how they might interact with a software system.

- **Initial Assumption:**

A receiver who needs blood goes by manual system of collecting blood and finds it too difficult to find a donor. Then, he comes to the contact by using this system. He accesses the system to find a blood donor.

- **Normal Scenario:**

The receiver accesses the system. At first, a list of donor appears before him/her. But he/she finds it too cumbersome to find a donor of his/her specific blood type. So, he/she goes to the search bar where he/she can search by blood group. He/she searches and finds some donors having the blood type that he/she needs. He/she clicks on a name and the donors profile appears before him/her. Now he/she wants to contact the donor. In the profile of the donor, he/she finds donor's contact information. Then he/she can send an e-mail

Now, the receiver wants to return the favor. He/she wants to act as a donor and donate blood to others. He/she returns to the homepage and finds a button named **Donor.** He/she clicks it and a log in page appears before him/her. But he/she haven't done registration. So, he/she decided to do registration first. He/she clicks on the **Registration** button and the registration page appears before him. He/she submits necessary information and clicks submit button. Then the system leaves a confirmation message of successful registration. Now he/she logs in. He/she notices that a profile has been created automatically with the information that he/she provides in the registration process. He/she also notices that he/she can update his/her profile. He/she also can access all the functions that he/she used to get as a receiver.

- **What can go wrong:**

The donor record does not exist or cannot be found. When the receiver search a blood group he/she could not found accurately or donor does not response timely.

- **Other activities:**

Users under 18 years old cannot use this system as a donor.

- **System state on completion:**

User will be logged off. Database will be closed and will be disconnected from the system.

2.5. Use Cases:

The e-blood system has basically two active actors and one cooperating actor admin. Admin activities will provide on the admin panel. The donor who has will to donate blood and the receiver who wants to collect blood. Each actor has ability to access his own activity functions in the system.

The first sections outlines the use cases for each of the active users(Donor & Receiver/patient). The final section outlines the use cases for the admin.

2.5.1.Use Cases for Donor and Receiver of the system:

UC1:Donor Registration

Actor: Donor

Preconditions:

Before this use case can be initiated, the user has already accessed the “e-blood” system .

Main success scenario:

1. The system opens the homepage.

2. The user requests to access as donor.

3. The system will open registration form.

6. The user will submit information accordingly.

7. The system will store the information in the database.

Post conditions:

1. System will show a confirmation message of completing registration.

Alternative Scenario:

Authorization Failed:

01. Prompt the user to be registered .

UC2:Receiver/Patient Registration

Actor: Receiver/Patient

Preconditions:

Before this use case can be initiated, the user has already accessed the “e-blood” system .

Main success scenario:

1. The system opens the homepage.

2. The user requests to access as receiver.

3. The system will open registration form.

6. The user will submit information accordingly.

7. The system will store the information in the database.

Post conditions:

2. System will show a confirmation message of completing registration.

Alternative Scenario:

Authorization Failed:

02. Prompt the user to be registered .

UC3:Login

Actor: Donor, Receiver/Patient

Preconditions:

1.Donor or receiver must be registered first in the e-blood system.

Main success scenario:

1. The system opens the homepage.

2. The user requests to access as a donor/receiver.

3. The system will open log-in page.

4. User will submit information.

5. The system will verify the information and return a Boolean as login status.

6.User view his/her profile information.

Post condition:

1. The system will show a confirmation message on successful Login.

UC4:Update_Donor_Info

Actor: Donor

Preconditions:

1. Donor has already accessed the system, completed registration or login process .

Main success scenario:

1. The system will show his/her account.

2. Donor will click the “Update information” button.

3. The system will allow the donor to update his/her information.

4. The donor will submit information.

5. The system will update the data stored in the server.

6. A confirmation message will appear.

UC4:Delete_Donor_ID

Actor: Donor

Preconditions:

1. Donor has already accessed the system, completed registration or login process.

Main success scenario:

1. The system will show his/her account.

2. Donor will click the “Delete ID” button.

3. The system will allow the donor to delete .

4. The system will delete the data stored form the server.5. A confirmation message will appear.

UC5:Display _Donor_ List

Actor: Donor, Receiver/Patient

Preconditions:

1.User have to visit on the homepage.

Main success scenario:

1. The donor will click to see the list of the donor.

2. The system will retrieve information from the server.

Post conditions:

1. The donors' list will be shown to the donor.

UC6:Search Donor by Blood Group

Actor: Donor, Receiver /Patient

Preconditions:

User have to visit on the homepage.

Main success scenario:

1. User will click to see the donors' list

2. The system will load the donors' list.

3. The donor will go to the search box and write down his/her desired blood group.

4. The system will show a list containing the donors' name with that specific blood group.

Post conditions:

1. The system will show the donors' list who have that specific blood type.

UC7:Search Donor by District

Actor: Donor, Receiver/Patient

Preconditions:

User have to visit on the homepage.

Main success scenario:

1. User will click "search donor by district" on the navigation bar.

2. The system will load the page.

3. The donor will go to the search box and write down his/her desired blood group and district .

4. The system will show a list containing the donors' name with that specific district..

Post conditions:

1. The system will show the donors' list who have that specific blood type.

UC8:Update_Receiver_Info

Actor: Receiver/Patient

Preconditions:

1. Receiver has already accessed the system, completed registration or login process .

Main success scenario:

1. The system will show his/her account.

2. Receiver will click the "Update information" button.

3. The system will allow the receiver to update his/her information.

4. The receiver will submit information.

5. The system will update the data stored in the server.

6. A confirmation message will appear.

UC9:Delete_Receiver_ID

Actor: Receiver/Patient

Preconditions:

1. Receiver has already accessed the system, completed registration or login process .

Main success scenario:

1. The system will show his/her account.

2. Receiver will click the "Delete ID" button.

3. The system will allow the receiver to delete .

4. The system will delete the data stored form the database system.

UC10:Display Receiver List

Actor: Donor, Receiver/Patient

Preconditions:

User have to visit on the homepage.

Main success scenario:

1. The user will click to see the list of the receiver.

2. The system will retrieve information from the database.

UC11:Display_Tips

Actor: Donor, Receiver/Patient

Preconditions:

User have to visit on the homepage.

Main success scenario:

1. The user will click to see the tips from the navigation bar.

2. The system will retrieve information from the database.

Post conditions:

1.System will show the Tips to the user**.**

UC12:Logout

Actor: Donor ,Receiver/Patient

Preconditions:

1.Actor must be logged in by his profile.

Main Scenario:

1. The system is running.
2. User wants to come out from the profile.
3. Click the” log out” button.
4. System displays successful message.

Post conditions:

1.System still stay on the home page.

2.User takes other facilities from the system.

Alternative scenario:

2a. User comes out from system without log out.

2a.01.Resume 1,2,3.

2.5.2. Use cases for admin of the system:

Brief Descriptions:

Admin is the one who's called site author or system administrator. He is a person responsible for maintaining all contents of the system. The duties of the admin may include: ensuring that the web servers, hardware and software are operating correctly, designing the system, generating and revising web pages. Admin can add donor & receiver also, can modify donor & receiver's details and can delete them too.

UC1:Login

Actor: Admin

Preconditions:

1.Admin does not need to register.

Main success scenario's:

1. The system will open log-in page.

2. admin will submit username and password.

3.System prompts the information.

3.Admin view his/her home page.

Post condition:

1. The system will show a confirmation message on successful Login.

UC2:Add_Donor

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the donor option.

2. admin will find add donor option and click it.

3.System will show a registration form.

4.admin will submit information accordingly.

3.System will store information in the database server.

Post condition:

1. System will show a confirmation message of completing registration.

UC3:Manage_Donor

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the donor option.

2. admin will find manage donor option and click it.

3.System will show the donor's list.

4.admin will delete/update according his choise.

5.Sytem will store information in the database server accurately.

Post condition:

1. System will show a confirmation message of completing delete/update of donor information.

UC4:Add_Receiver

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the receiver option.

2. admin will find add receiver option and click it.

3.System will show a registration form.

5.admin will submit information accordingly.

3.System will store information in the database server.

UC5:Manage_Receiver

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the receiver option.

2. admin will find manage receiver option and click it.

3.System will show the receiver's list.

4.admin will delete/update the receiver information according to his choice .

5.Sytem will store information in the database server accurately.

Post condition:

1. System will show a confirmation message of completing delete/update of donor information.

UC6:Add Health Tips

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the heath tips option.

2. admin will find add health tips option and click it.

3.System will show a insert health tips form.

4.admin will submit information accordingly.

3.System will store information in the database server.

Post condition:

1. System will show a confirmation message of completing delete/update of features information.

UC7:Manage_Health_Tips

Actor: Admin

Preconditions:

1.Admin must stay on admin panel home page..

Main success scenario's:

1. admin click on the health tips option.

2. admin will find "manage health tips" option and click it.

3.System will show the health tips list.

4.admin will delete/update the tips information according to his choice .

5.Sytem will store information in the database server accurately.

Post condition:

1. System will show a confirmation message of completing delete/update of donor information.

UC8:Logout

Actor: Admin

Preconditions:

Actor must be logged in admin panel.

Main success scenario's:

1. The system is running.
2. User wants to come out from the panel.
3. Click the" log out" button.
4. System displays successful message.

Post conditions :

System displays home page.

Alternative scenario:

2a. User comes out from system without log out.

2a.01.Resume 1,2,3.

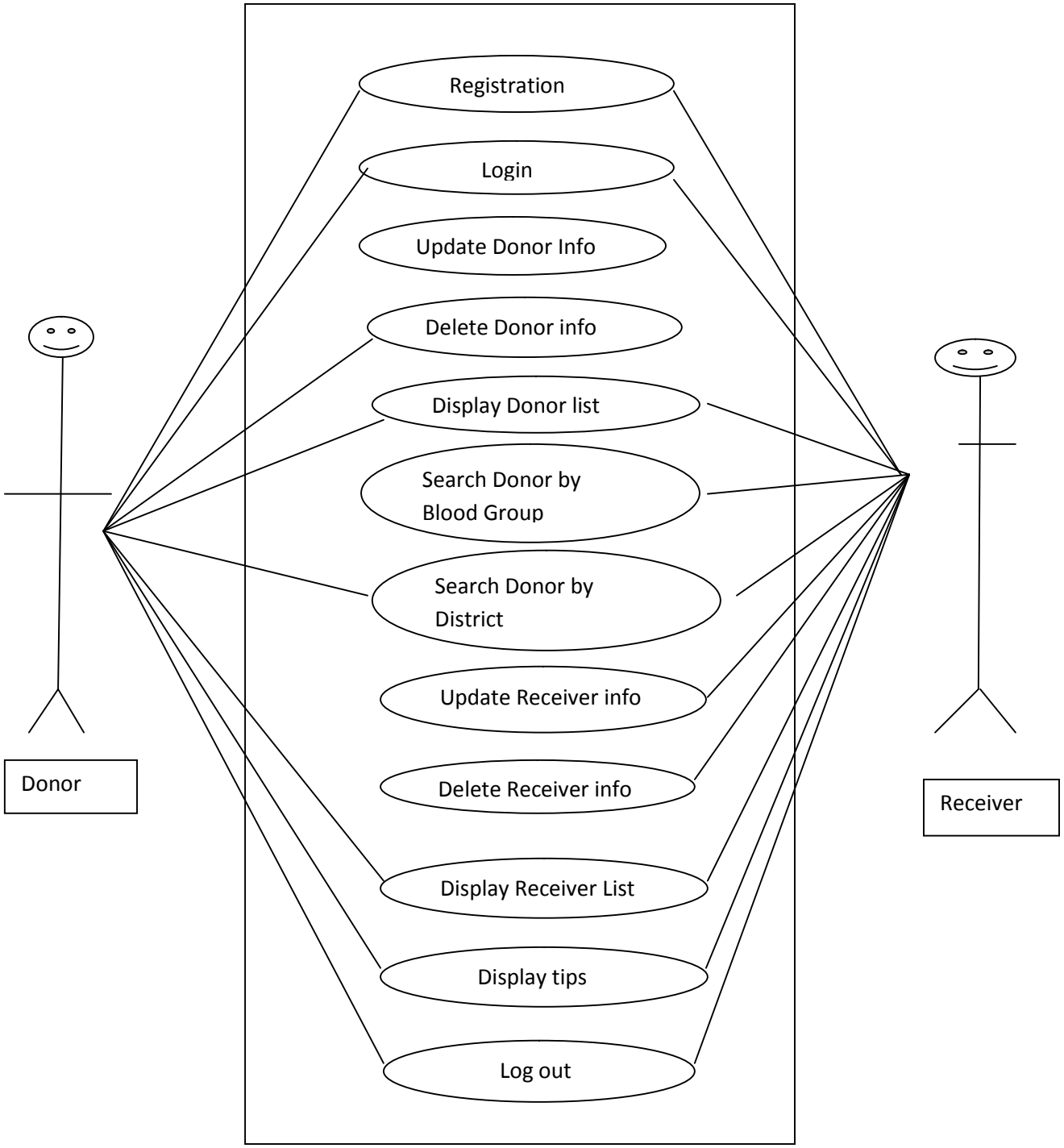
2.6.Use Case Diagram:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases an will often be accomplished by other types of diagram as well.

It identifies actor which are external entities that interact with the system. This view represents the system from the users perspective. The analysis on the above representation describes a usage scenario from the end-users perspective.

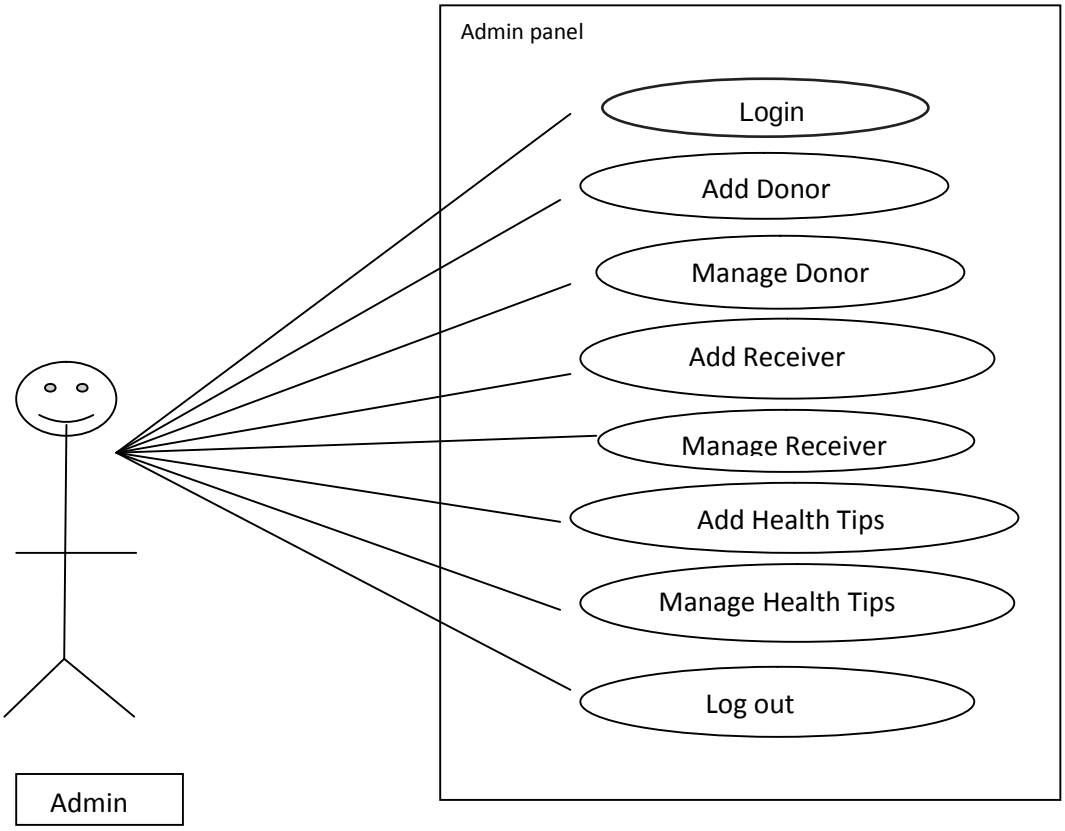
2.6.1.Use Case for Donor & Receiver Modules:

According to the system regarding use cases of the system is illustrated below:



2.6.2.Use Case for Admin Module:

This model the data and functionality are arrived from inside the system..This model view models dynamic structures and it describes the system scenario's of admin.



2.7.DFD of the system:

Data flow diagram **(DFD)** is one of the major graphical structure analysis modeling tools that allows us to depict a system as a network of functional processes, connected to one another. **DFDs** can't only be used modeling information processing system, but also as a way of modeling whole organization.

The components of **DFD** are the **Process**, the **Flow**, the **Store** and the External Entity. DFD illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from ,where its goes and how it gets stored. The DFD of this case is not perfect in my introduction chapter. So In this section, we include the DFD of e-blood system :

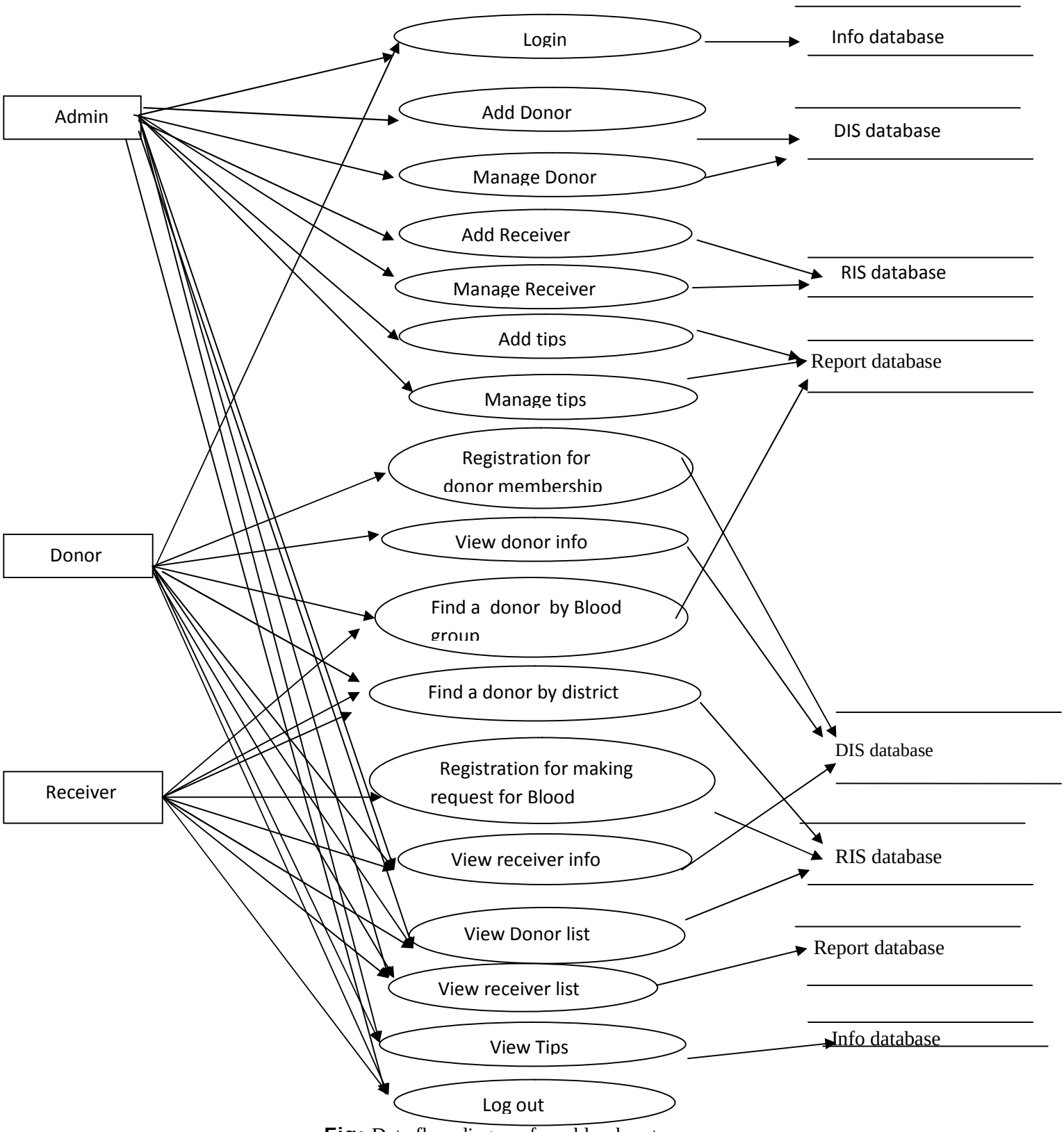


Fig: Data flow diagram for e-blood system

2.8.Conclusion:

A good SRS documents specifies all the features required in the final system. It is used by the customer to determine whether all the features have been provided in the delivered software system. It helps to accurately estimate the amount of effort required. Since SRS precisely defines the project scope, it ensures that customer's expectation don't change during software development.

CHAPTER 3:DETAILED DESIGN DOCUMENT

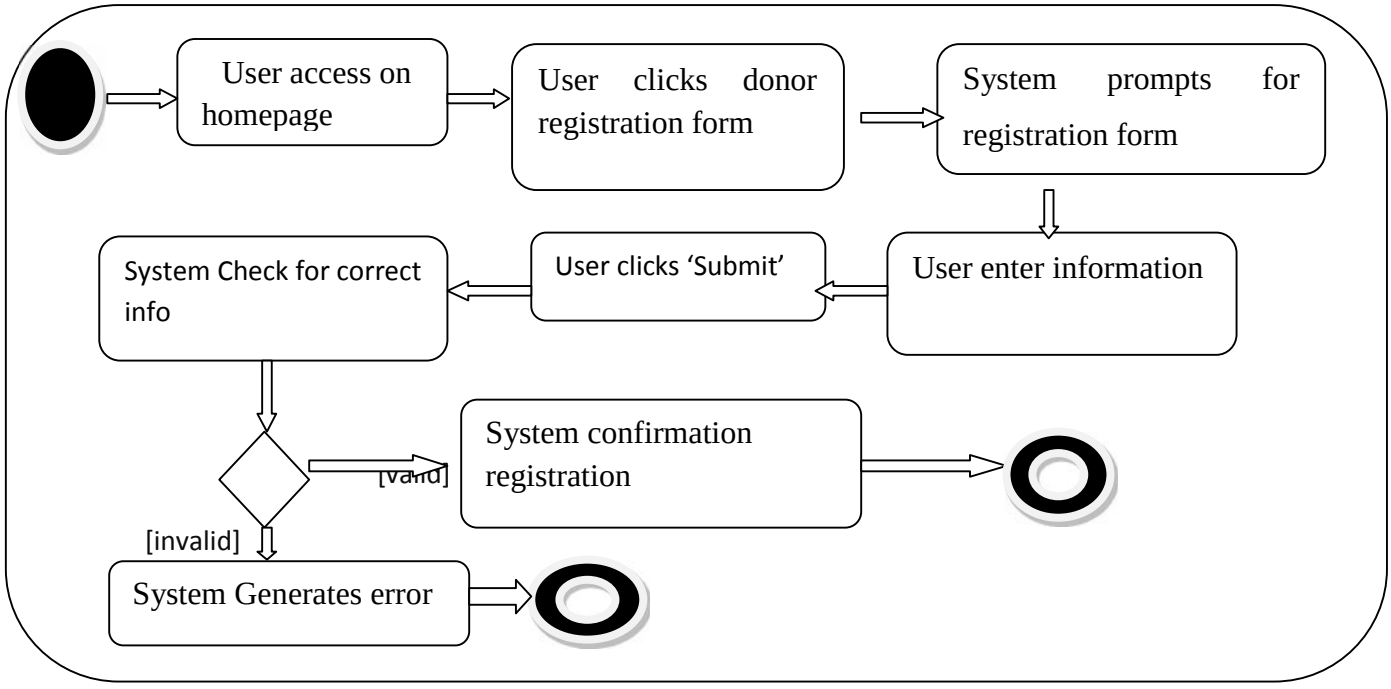
e-blood system is an Web based application which helps in donating and receiving blood. The objective of this document is to provide a detailed design description of “e-Blood” system. I have included Activity diagram, Class diagram and Sequence Diagram to analyze the system from different perspective and ease the system implementation. Each model in this document is represented based on graphical notation using UML.The topics that have discussed here are stated below:

- Activity diagram.
- Class diagram.
- Sequence diagram.
- Architectural pattern.

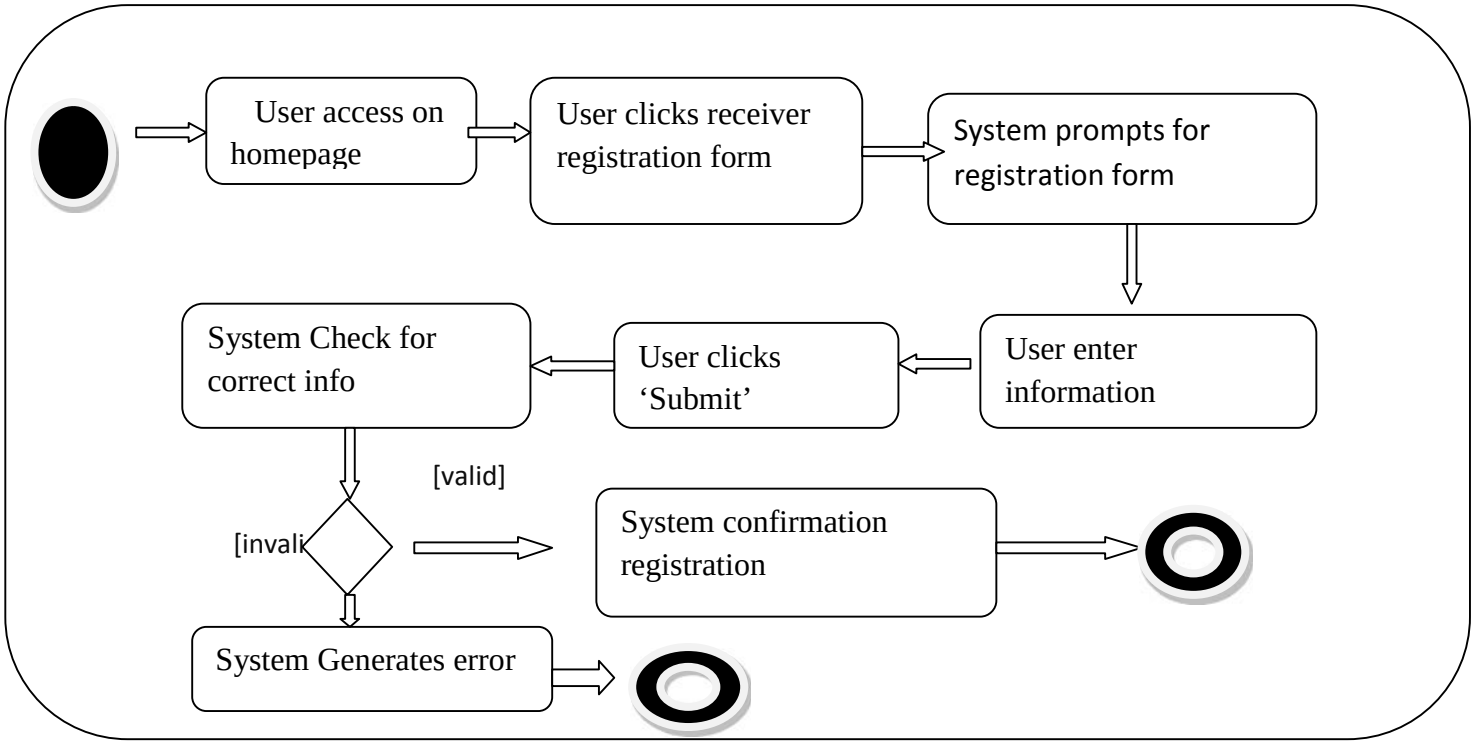
3.1. Activity Diagram: The UML diagram called Activity Diagram specifies the actions and the flow among those actions of each Use Case specifies in SRS document. First of all, The EBS has following use cases for donor & receiver are : Registration ,Login, update donor info , delete donor info, Display donor list,Search by blood group, update receiver info, delete receiver info, display receiver list, logout.

3.1.1.Activity Diagram for Donor & Receiver:

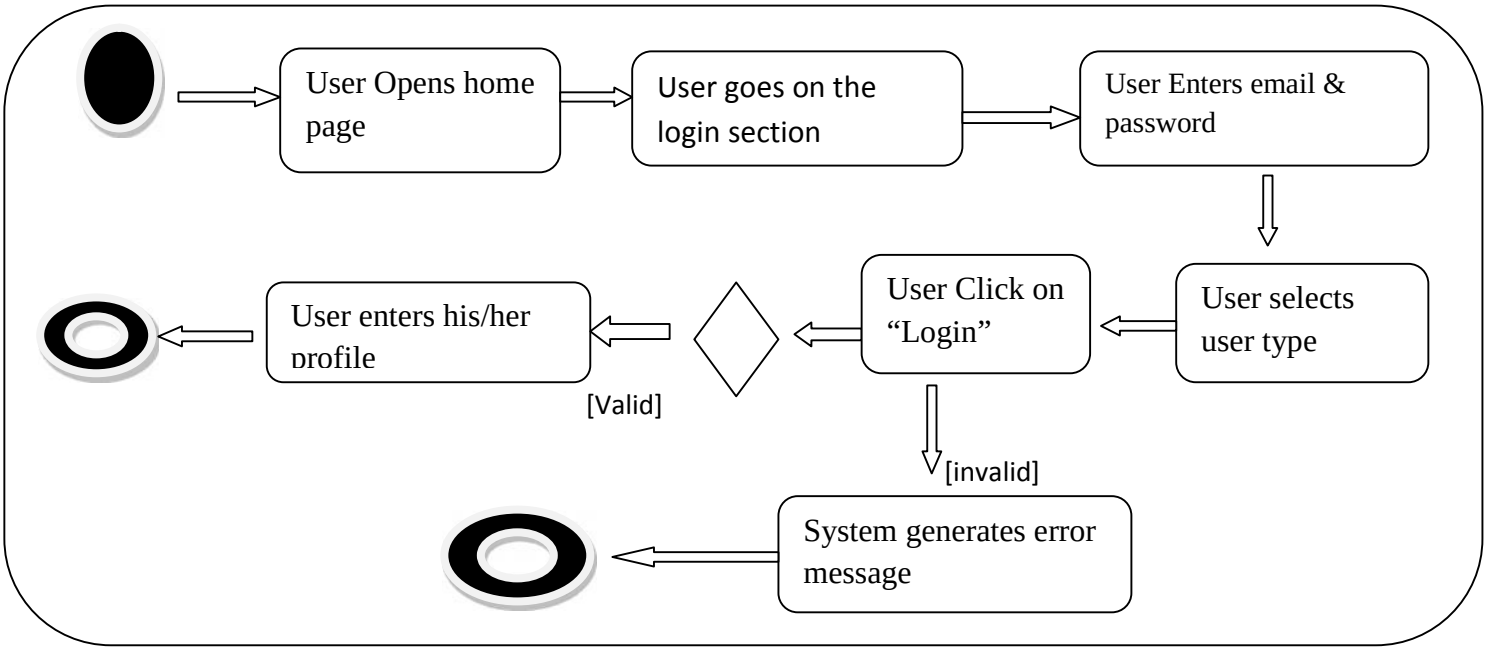
A.Donor Registration:



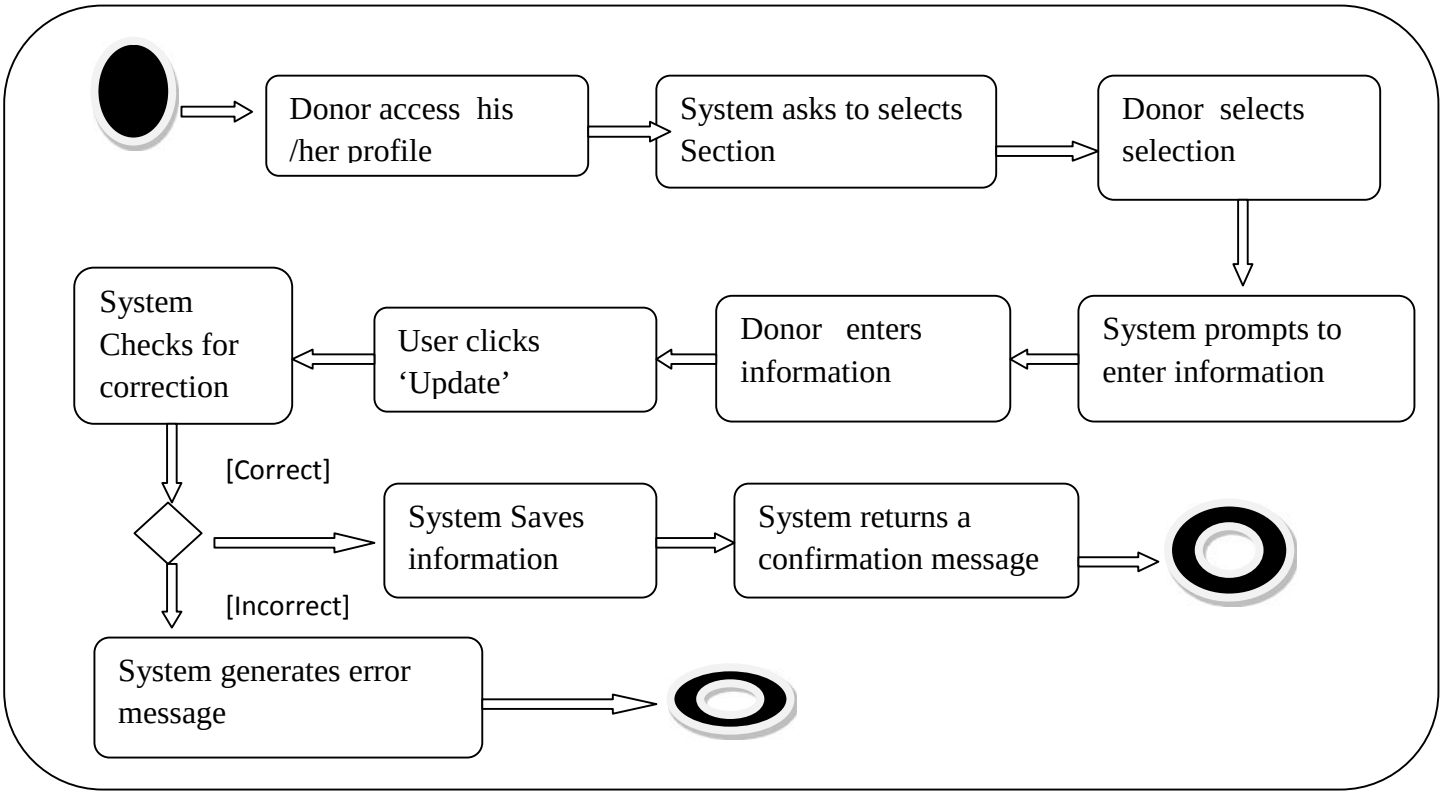
B.Patient/Receiver Registration:



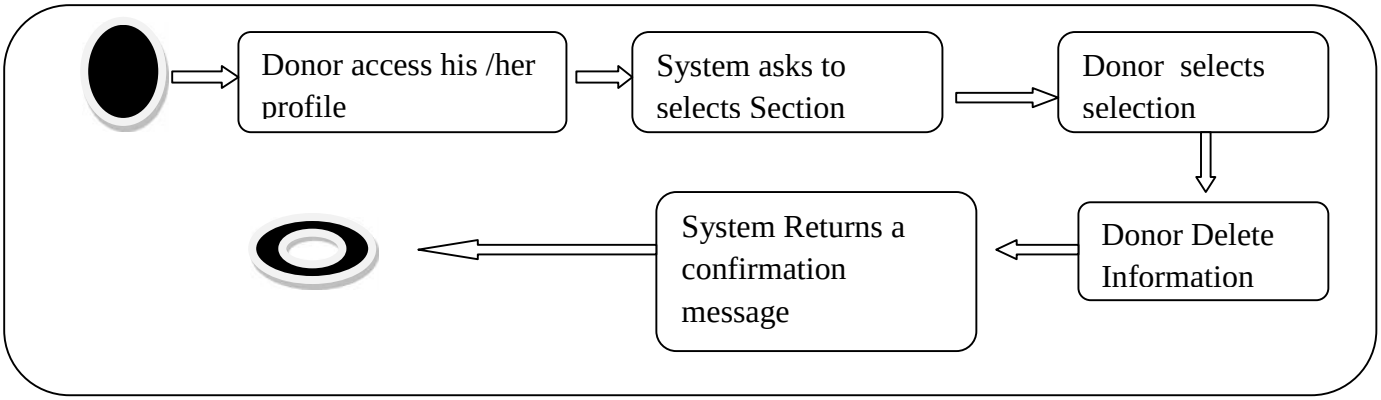
C.Login:



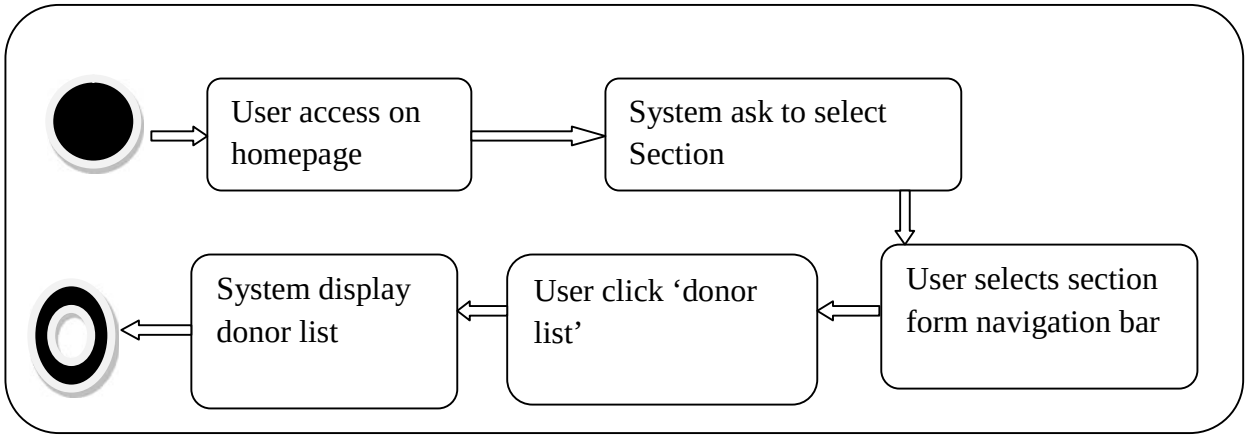
D.Update Donor Information:



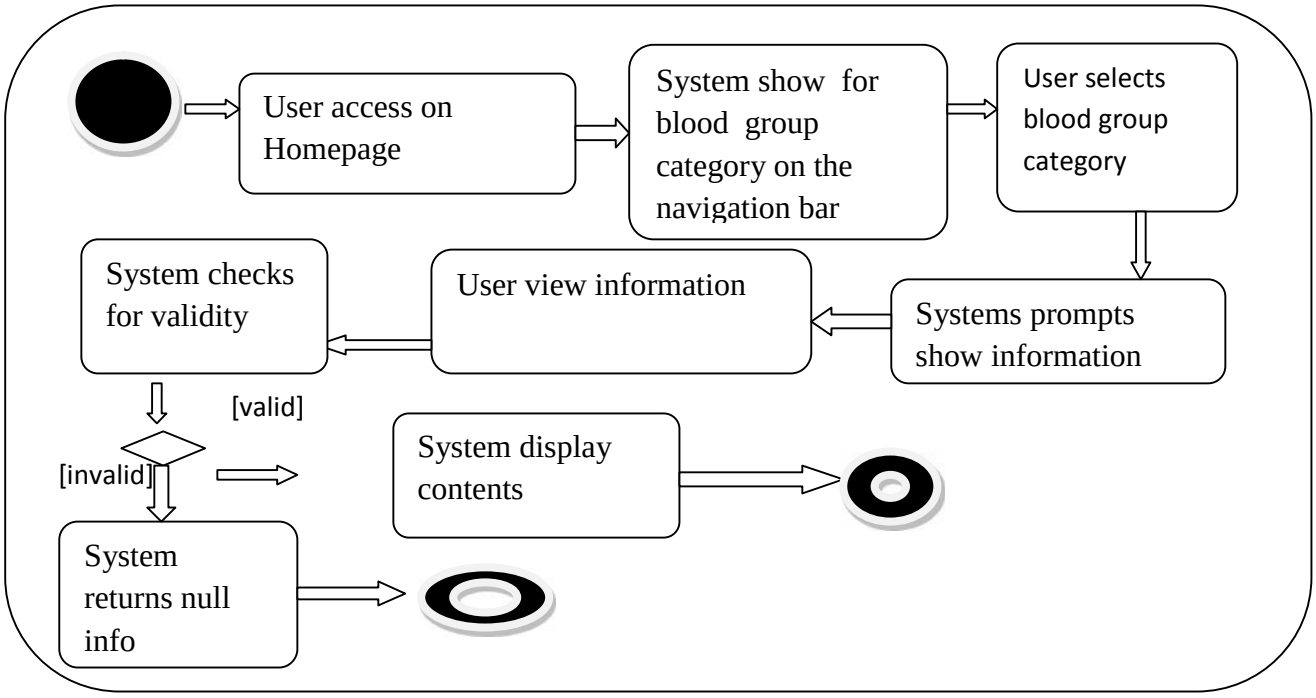
E.Delete Donor Information:



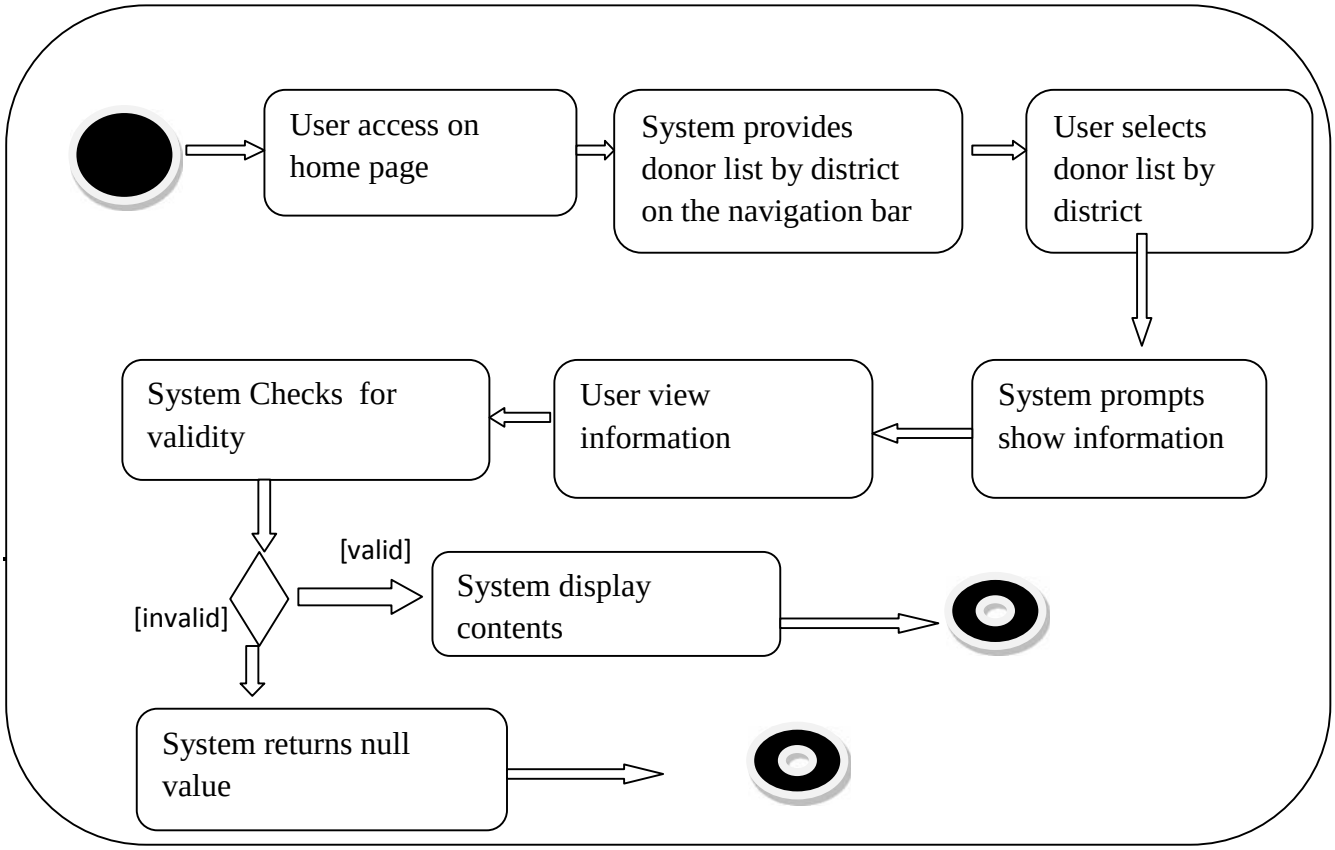
F.Display Donor List:



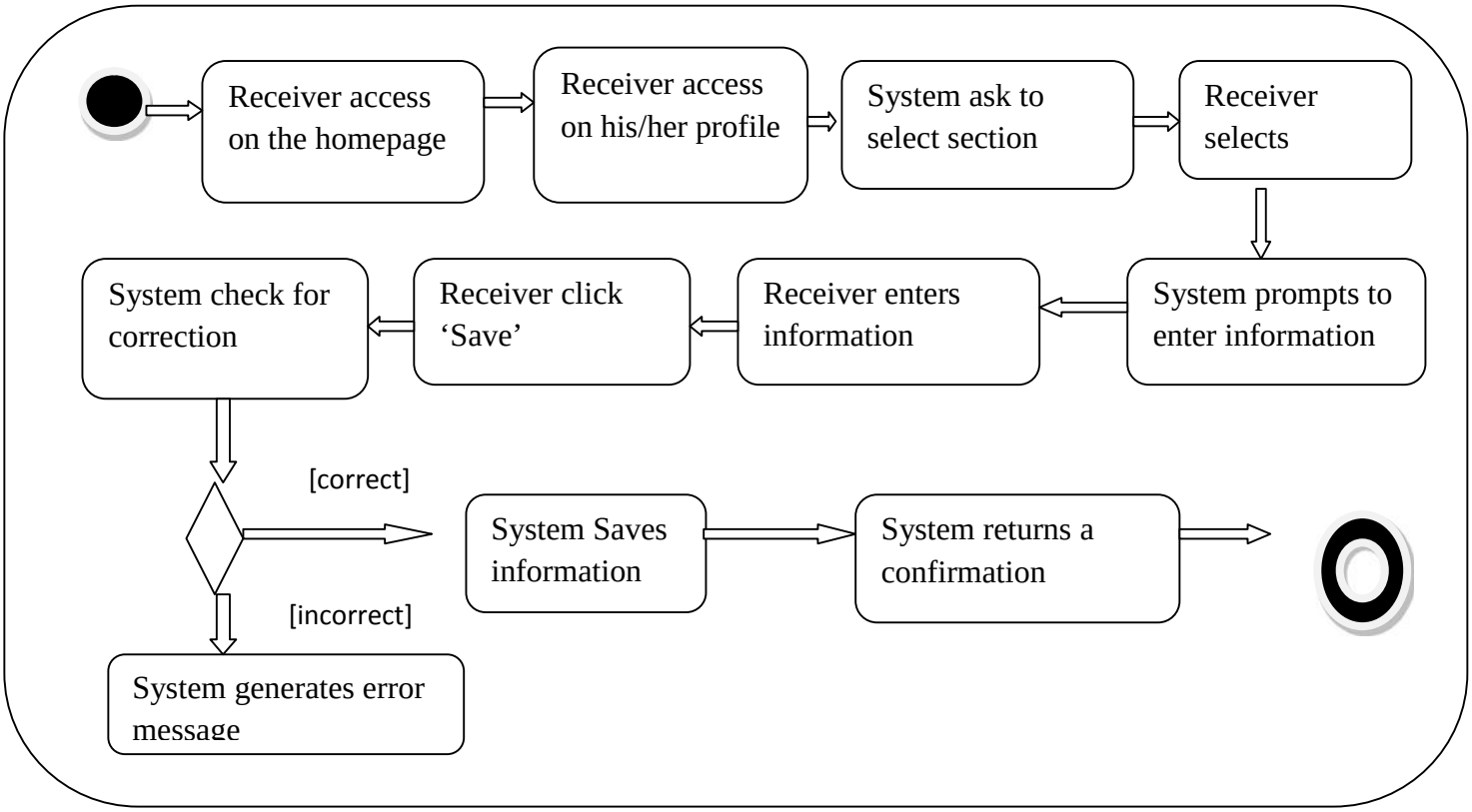
G.Search Donor by Blood group:



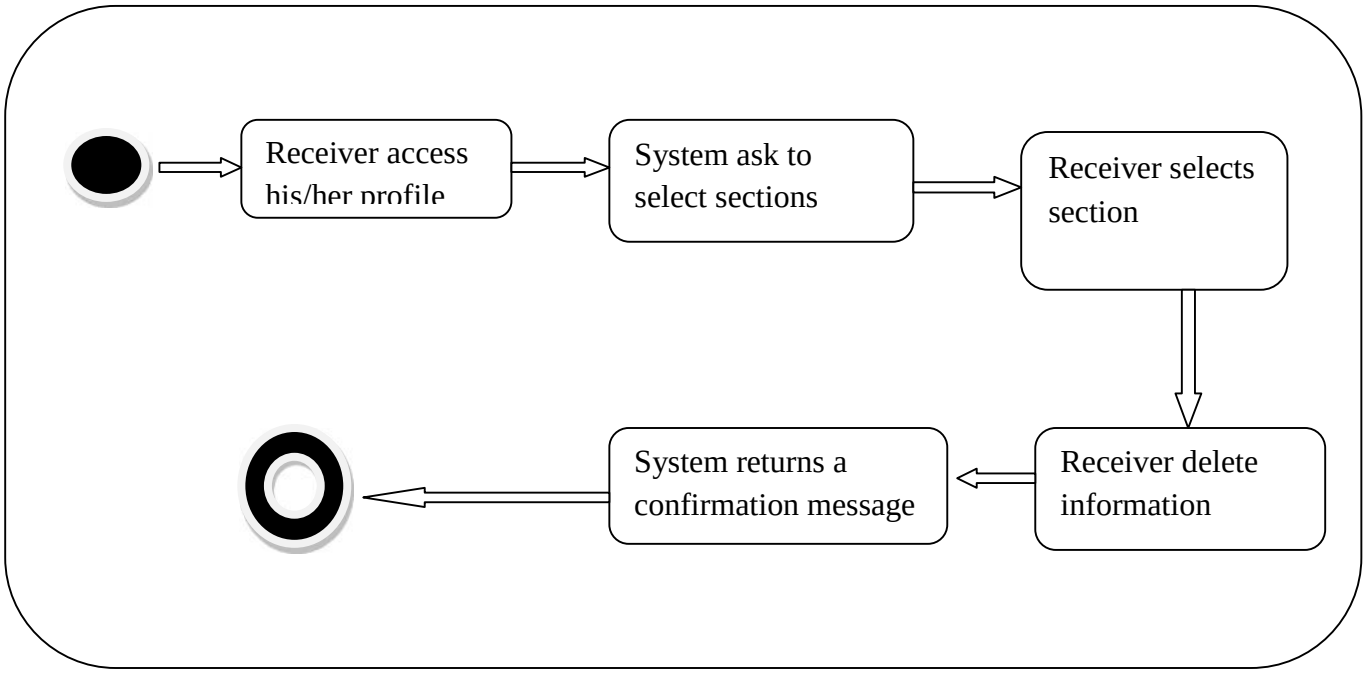
H.Search Donor by District:



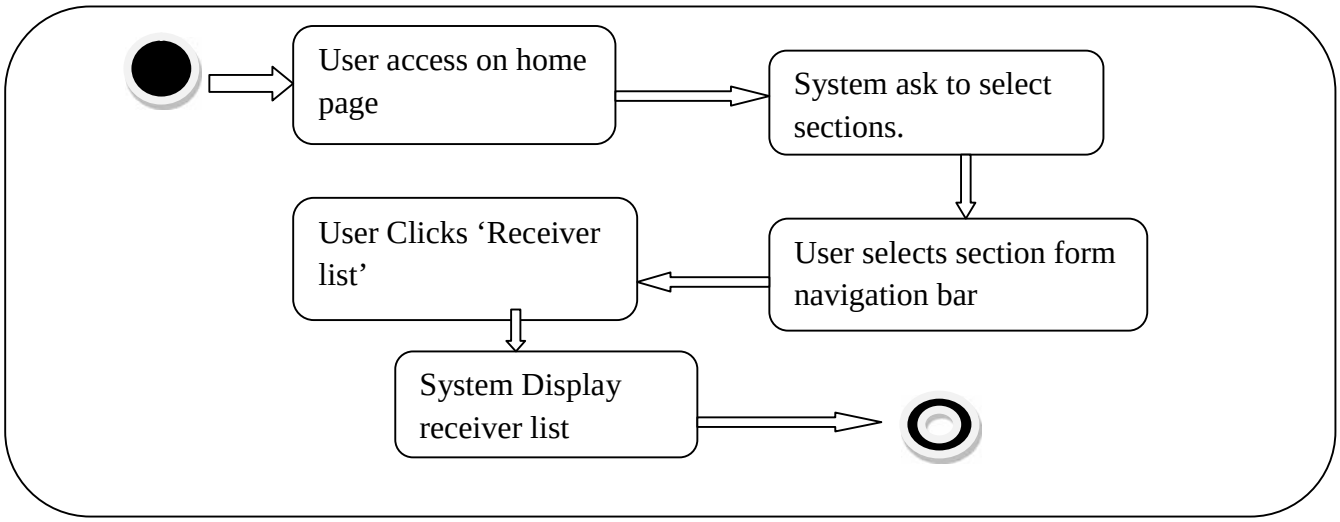
I.Update Receiver Information:



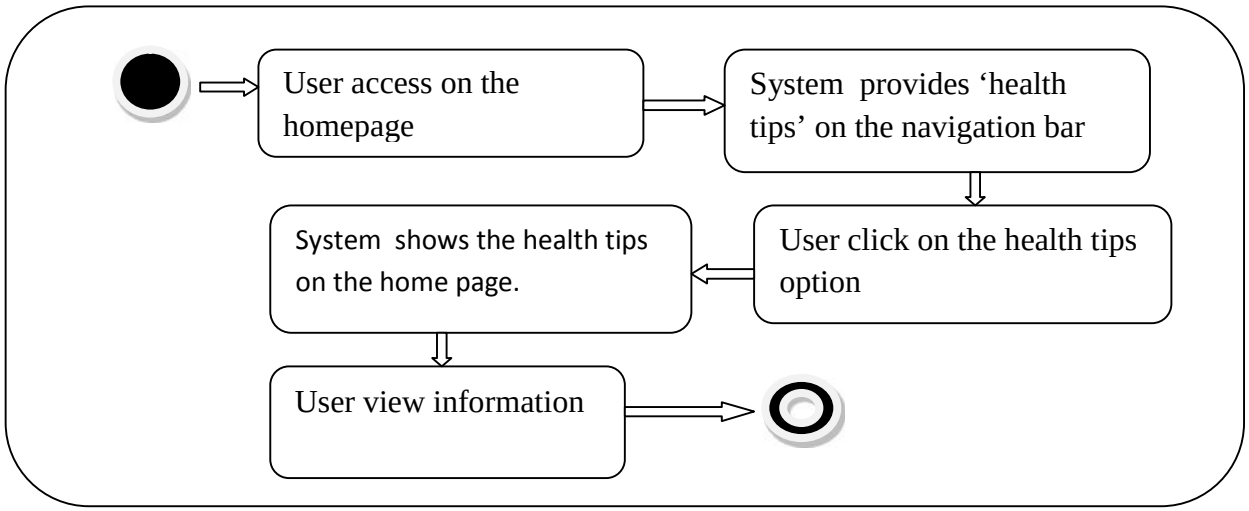
J.Delete Receiver Information:



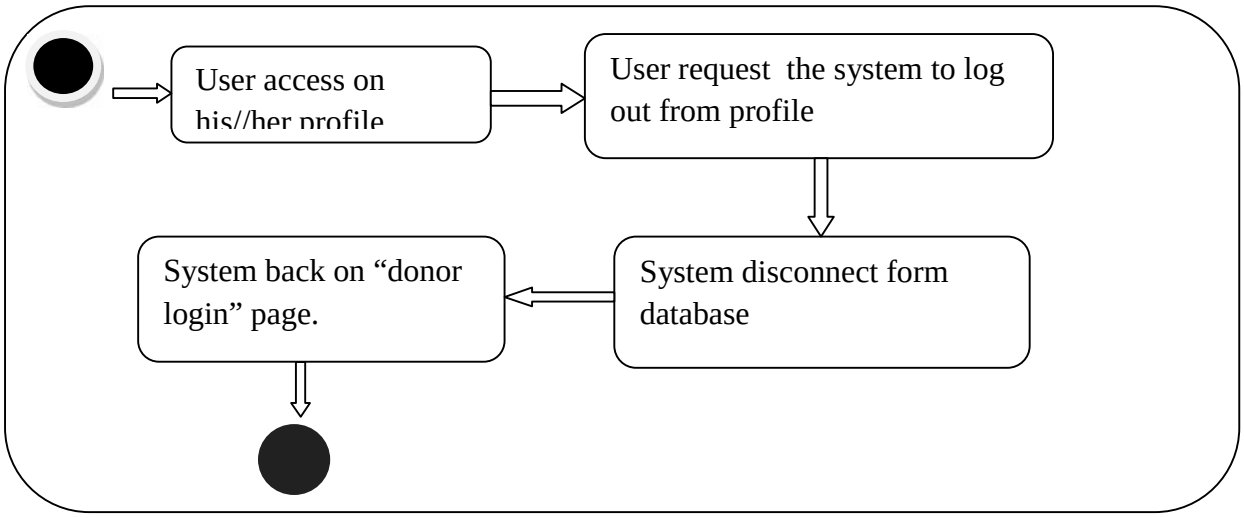
K.Display Receiver List:



L.Display Health Tips:

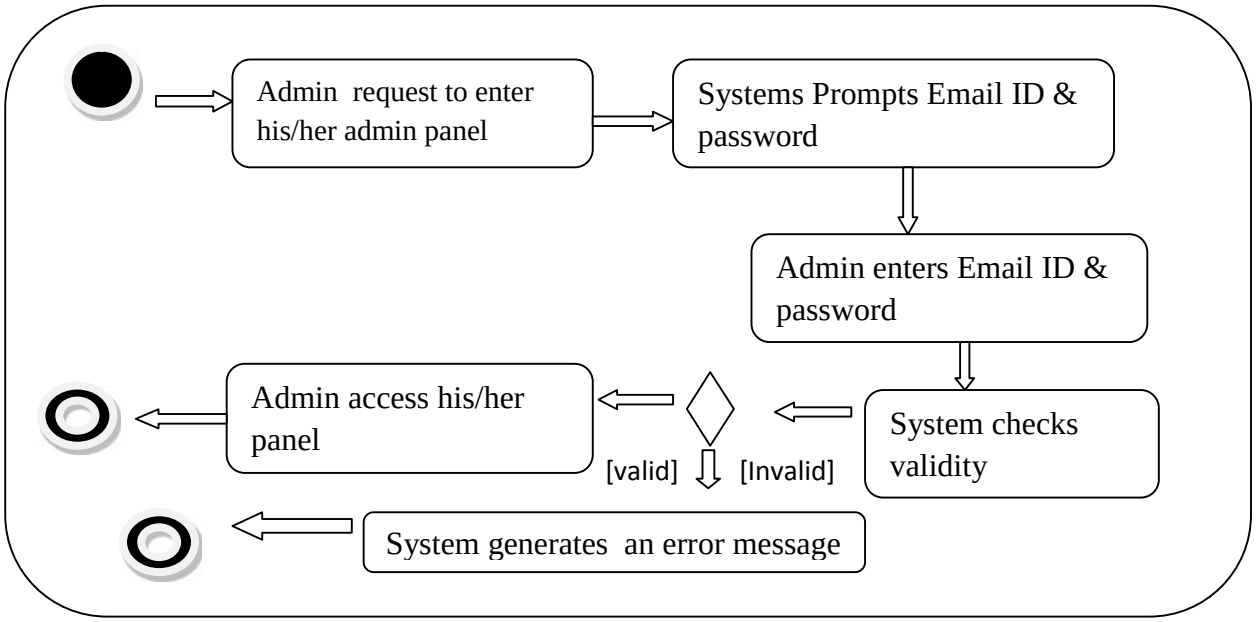


M.Log Out:

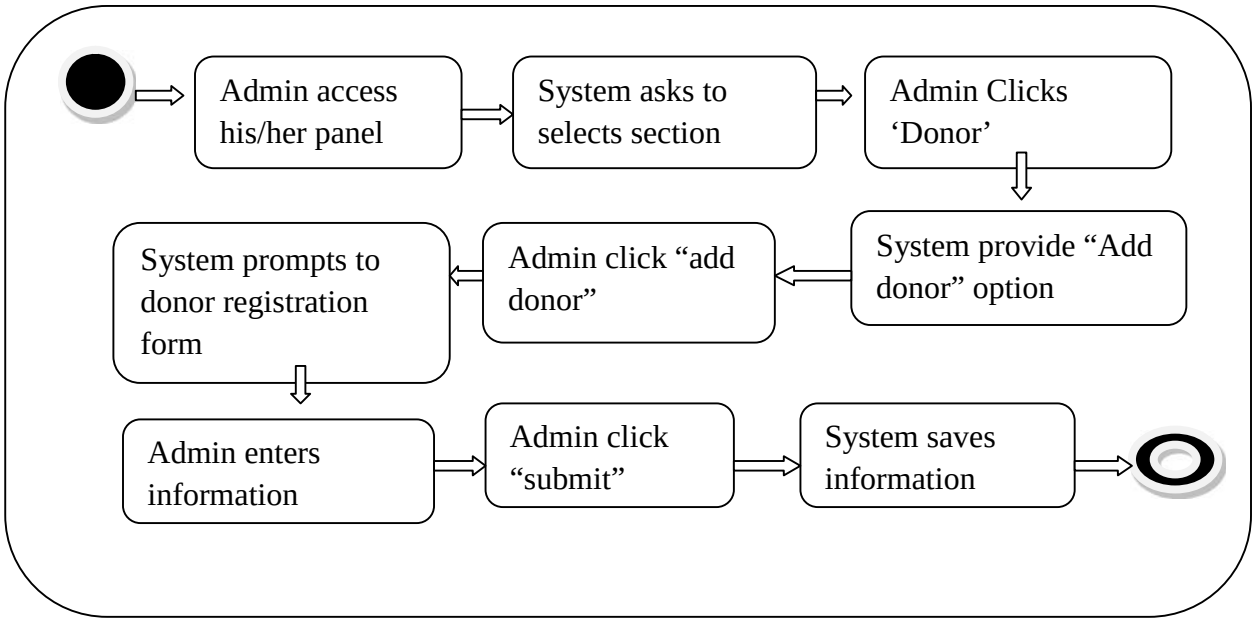


3.1.2.Activity Diagram for Admin:

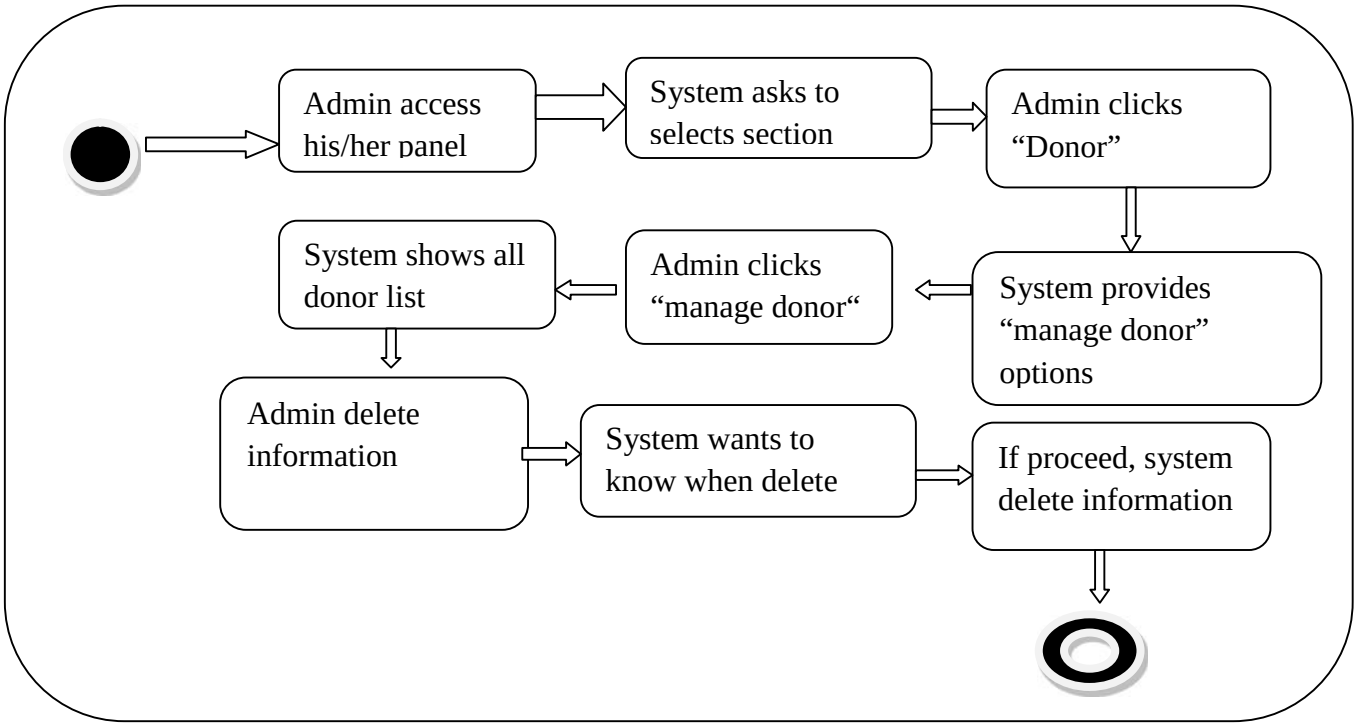
A.Admin Login:



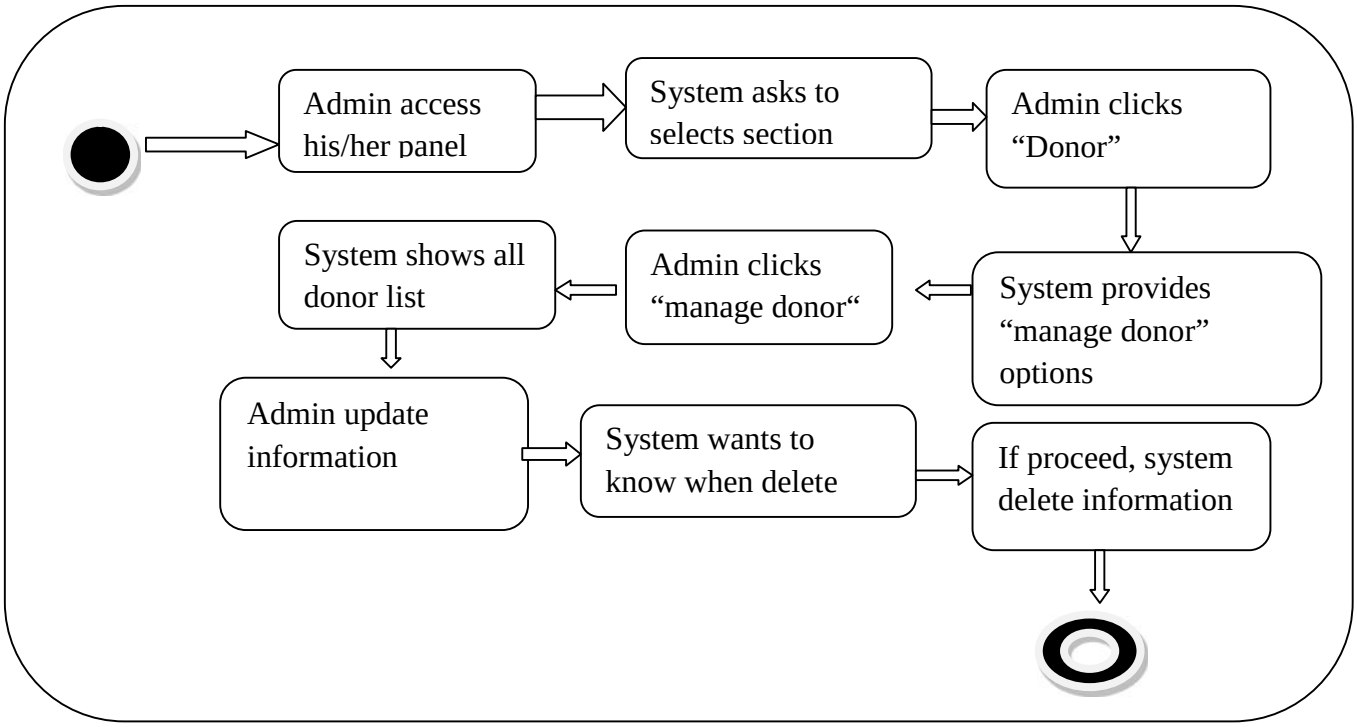
B.Add Donor:



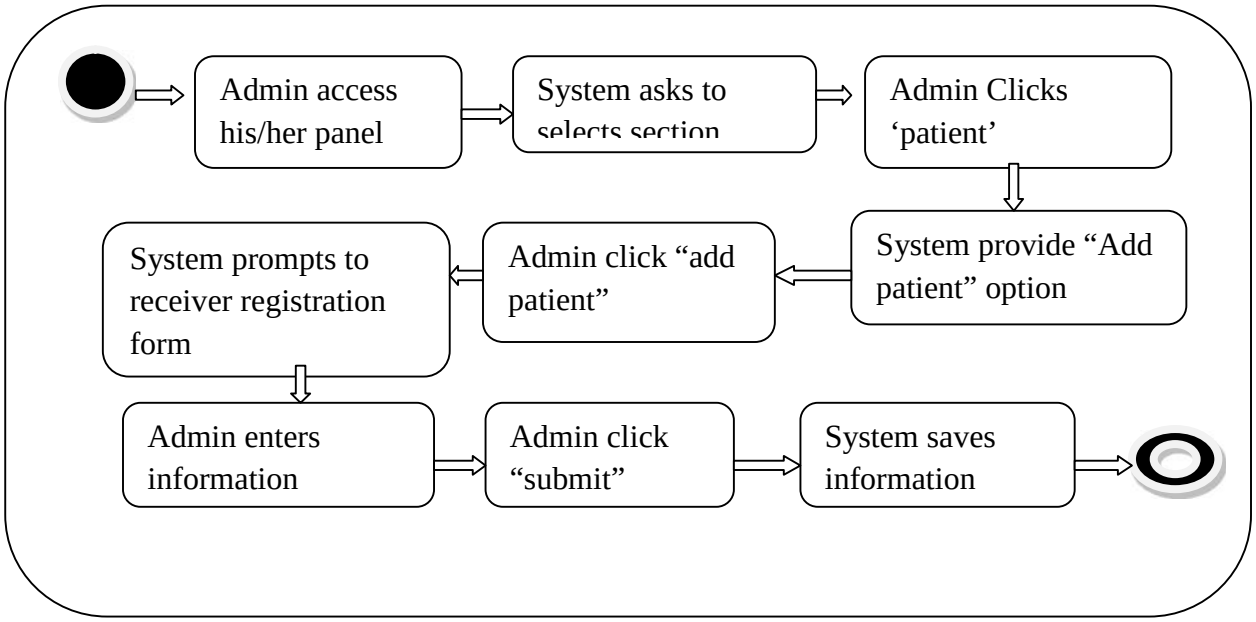
C.Delete Donor:



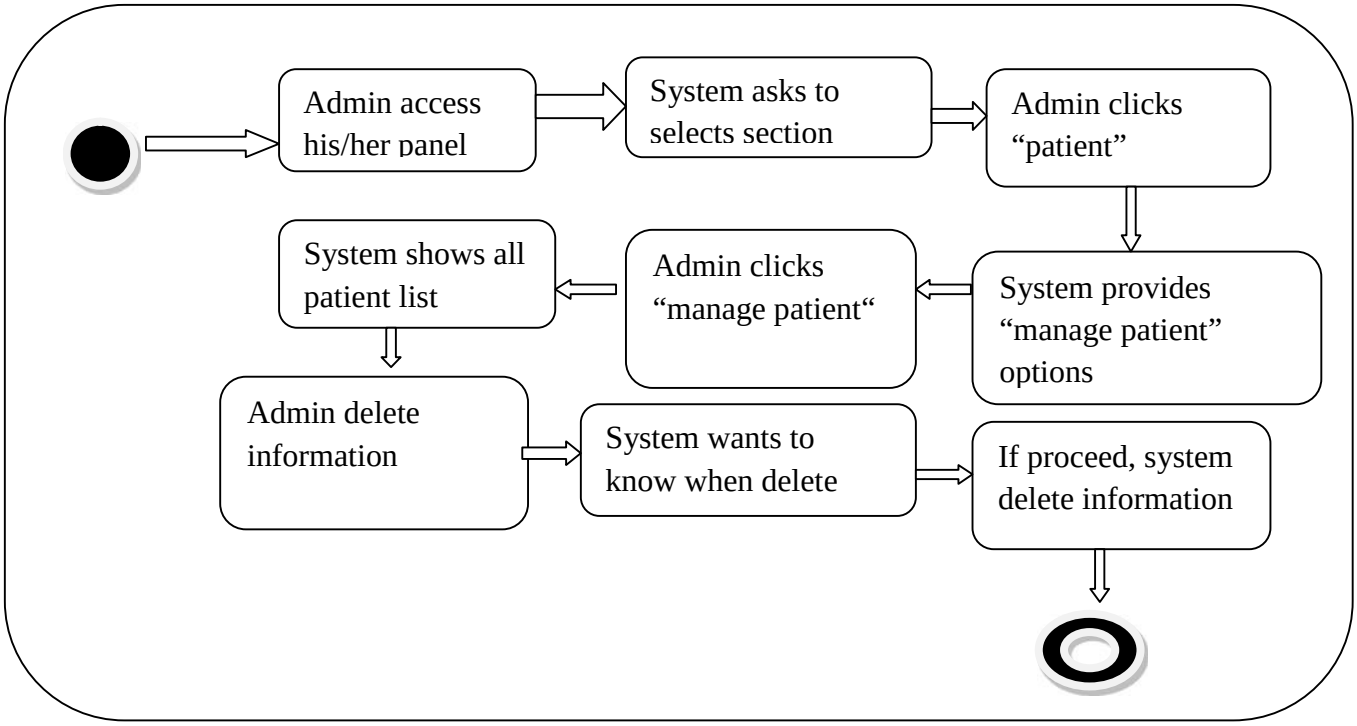
D.Update Donor:



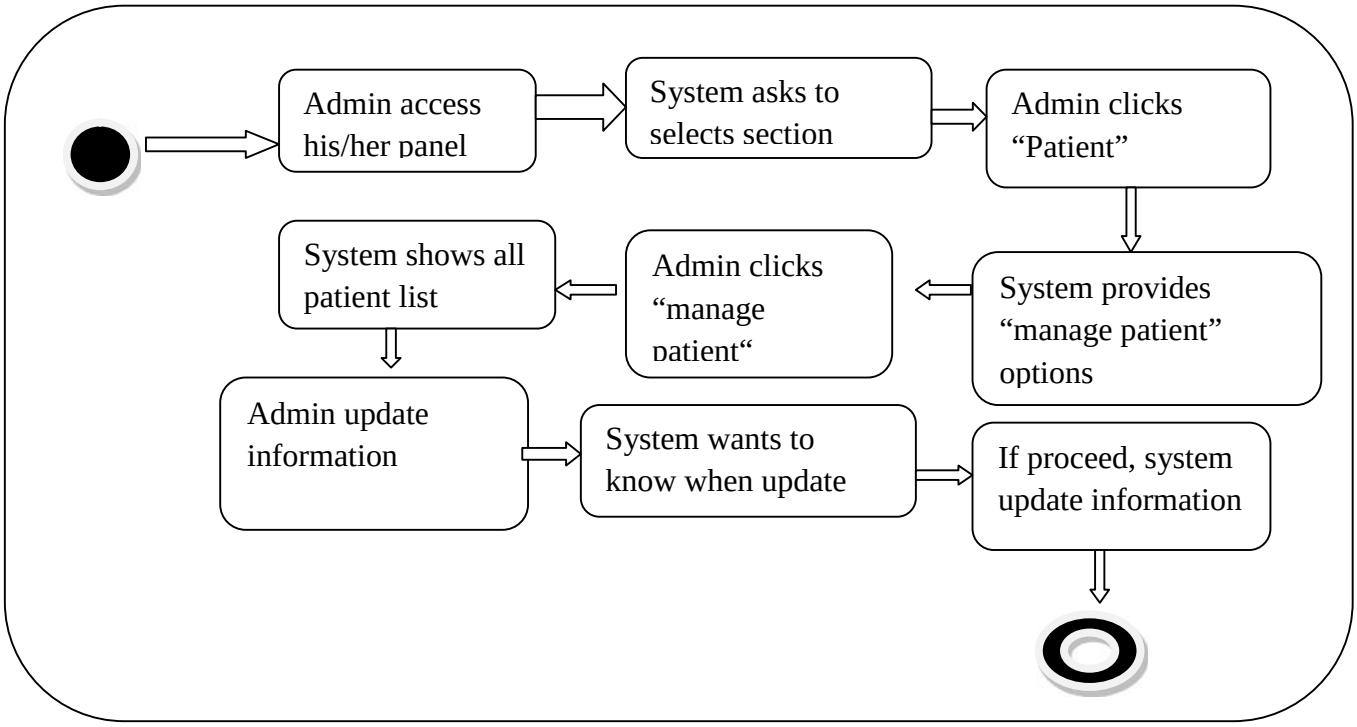
D.Add Receiver/Patient:



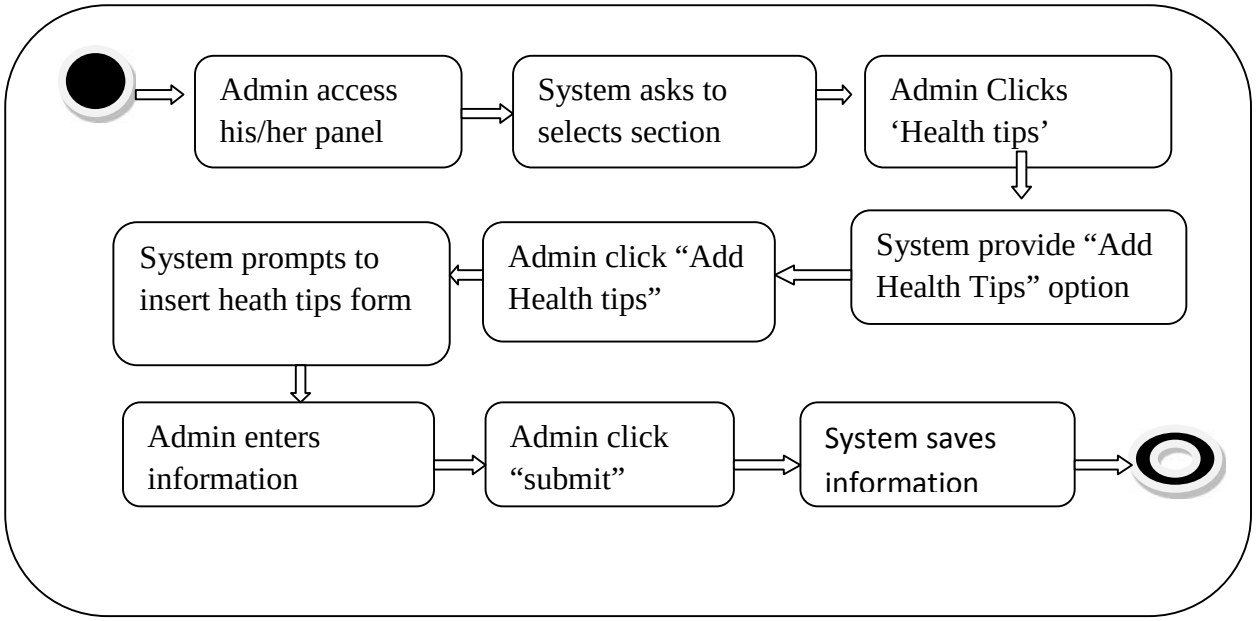
C.Delete Receiver/Patienr:



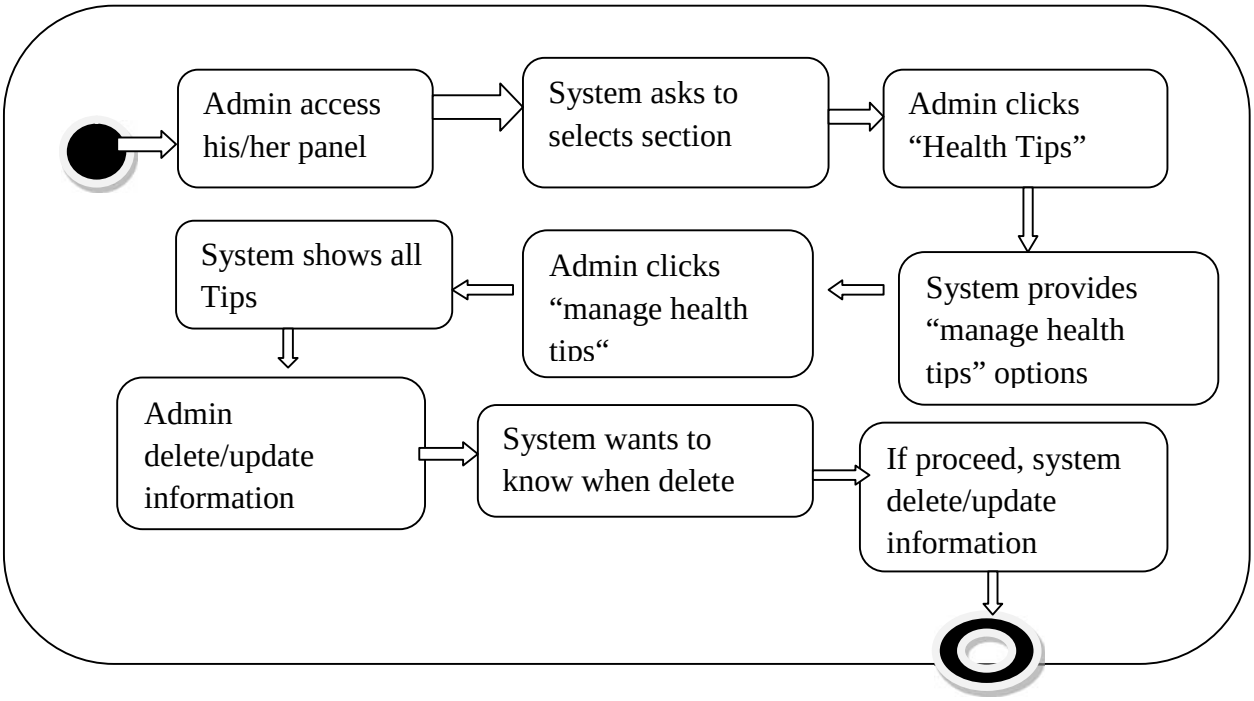
D.Update Patient:



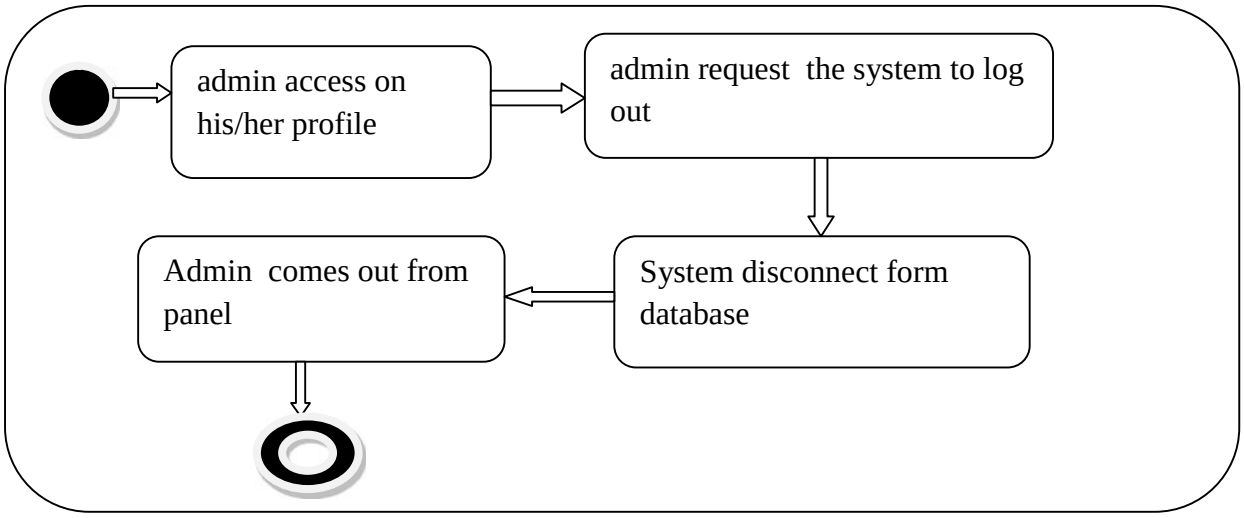
E.Add Tips:



F.Manage Health Tips:



G.Log Out:



3.2.Conceptual Class Diagram:

Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes.It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system.The class diagram of e-blood system for Donor and receiver/patient with identified class, methods and attributes are shown in below:

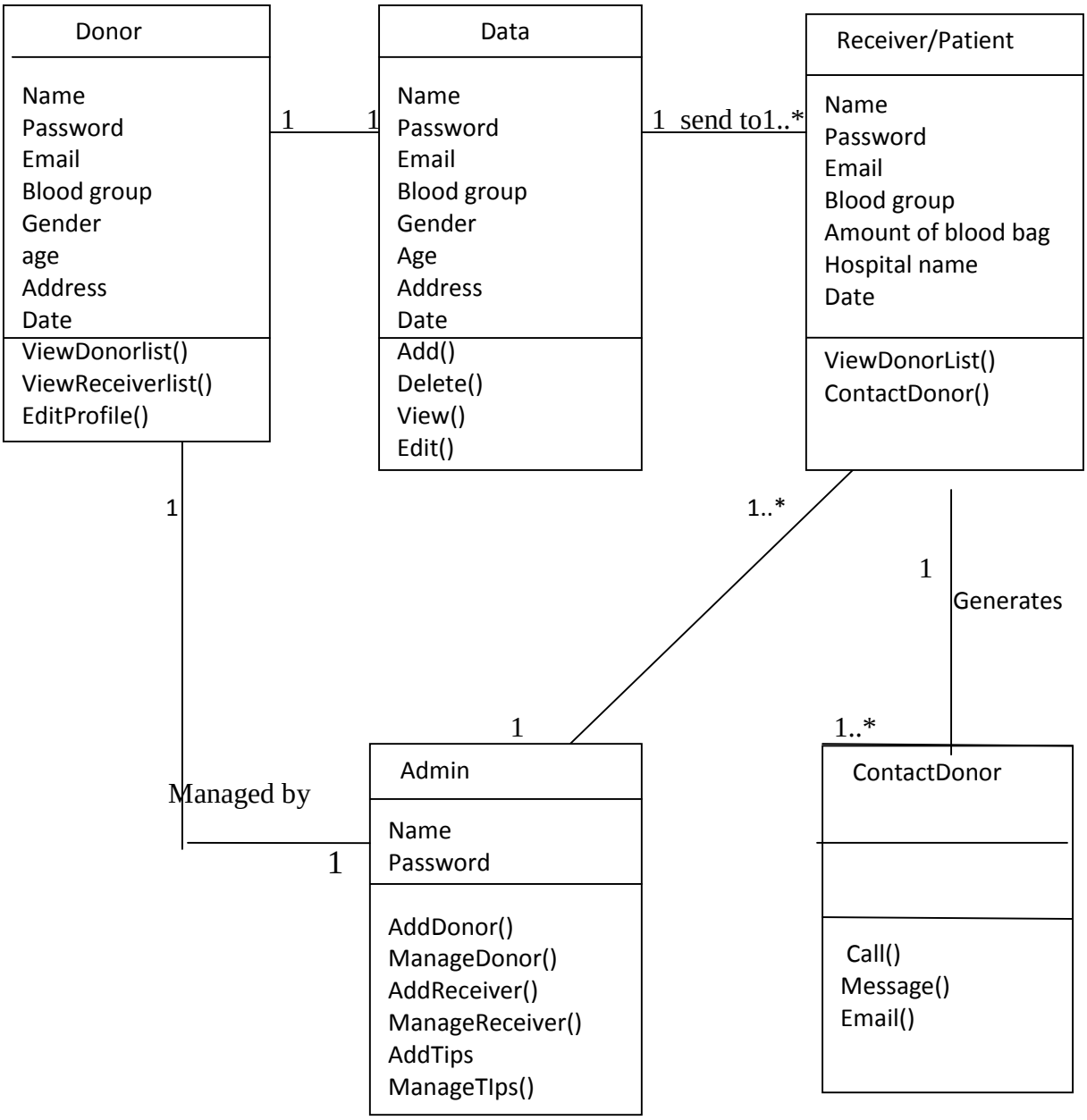


Figure :Conceptual Diagram For e-blood system

3.3. Sequence Diagram

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interaction occur.The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. Diagram specifies the interaction of objects for each use case. The sequence diagrams will communicate what messages are sent between a system's objects as the order in which messages occur. The diagram convey this information along the horizontal and vertical dimensions. First of all here The Donor & Receiver Sequence diagram are given below:

3.3.1. Sequence Diagram for Donor & Receiver:

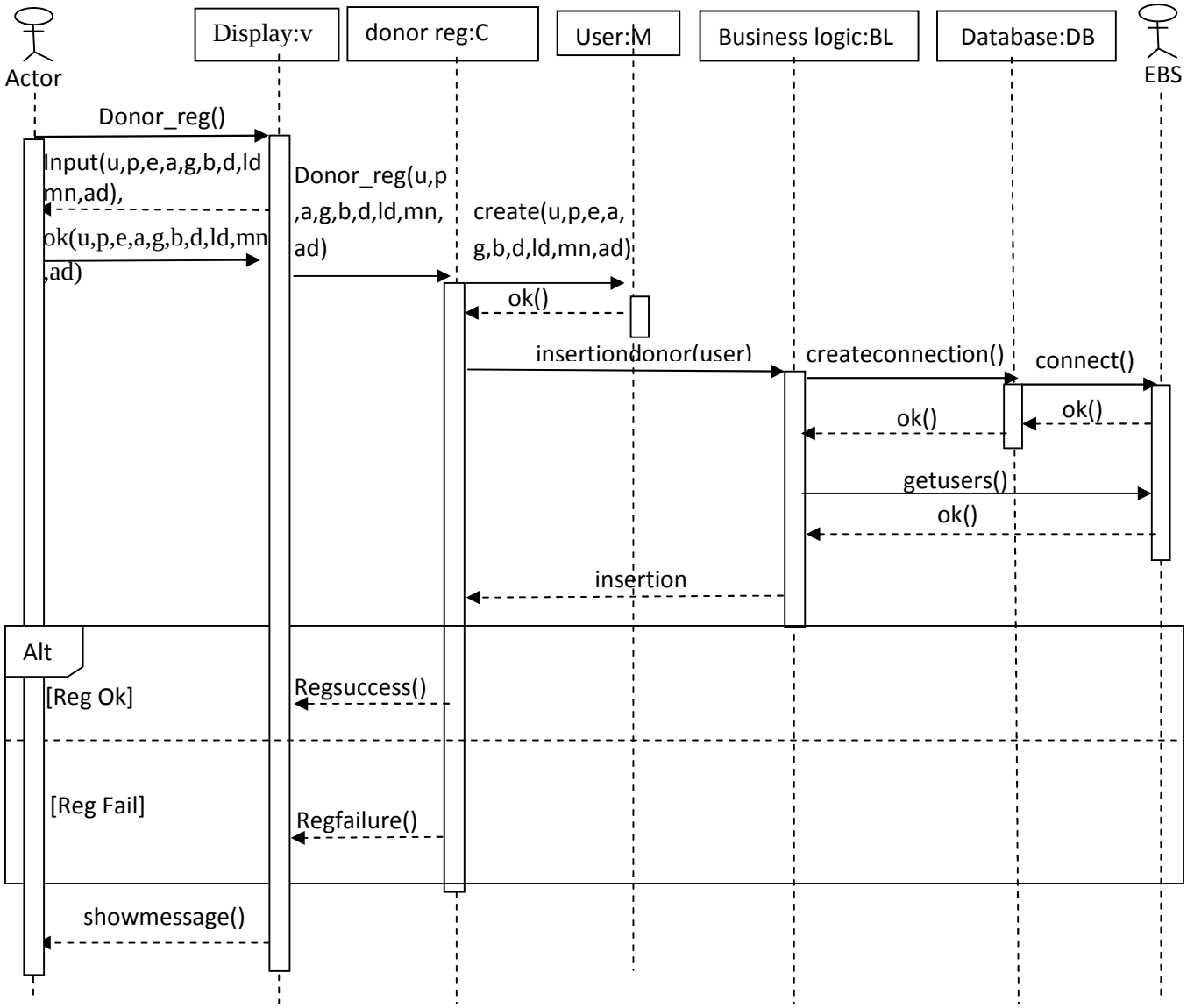


Figure 1:Sequence diagram of Donor registration form

Here,u=user name,p=password,e=Email,a=Age,g=Gender,b=blood group,d=district,ld=last donated date,mobile no,ad=address

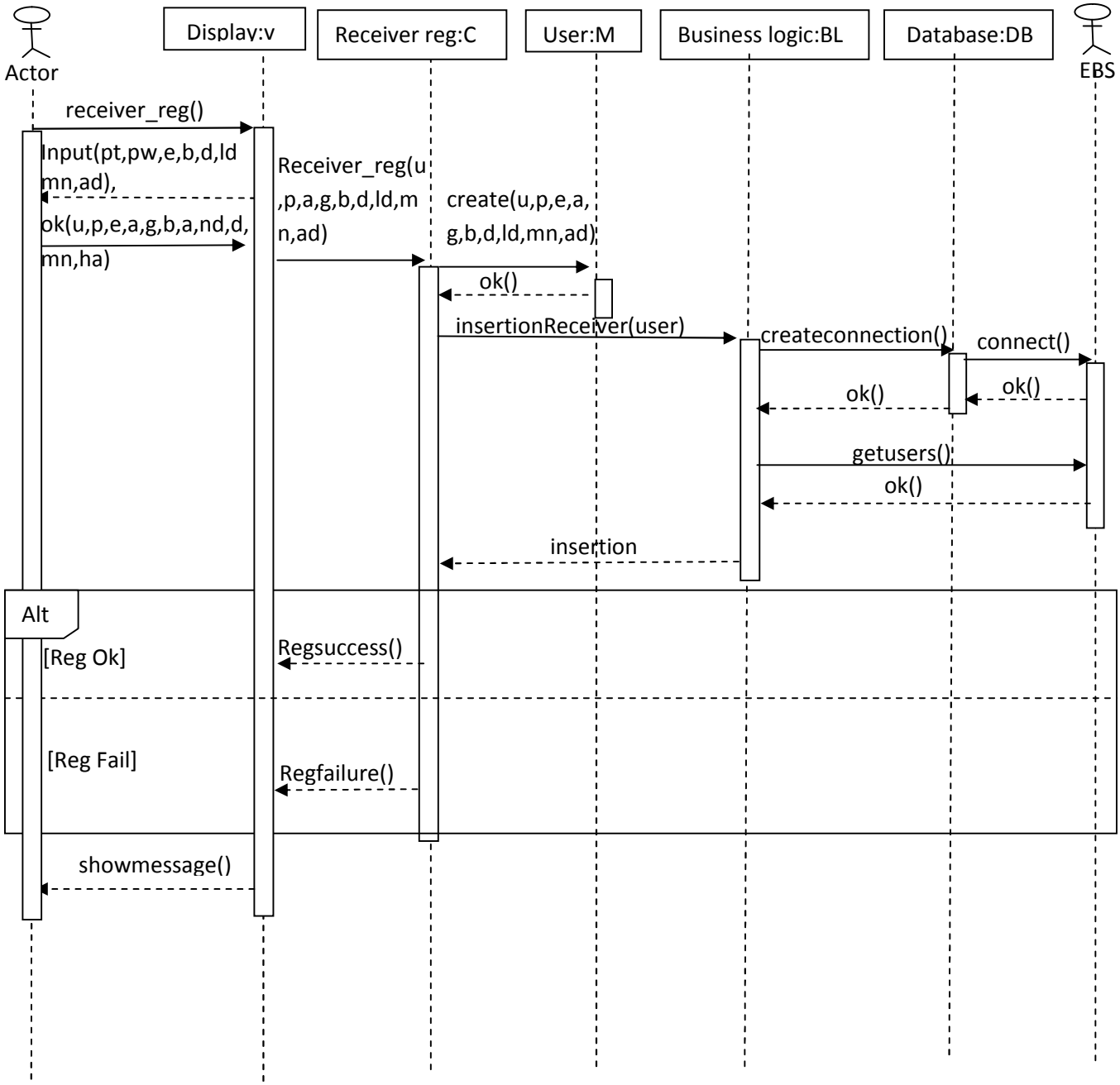


Figure 2:Sequence diagram of Receiver registration form

Here, Pt=patient name,pw=password,e=email,b=blood group,d=district,ld=last dated,mn=mobile no,ad=address

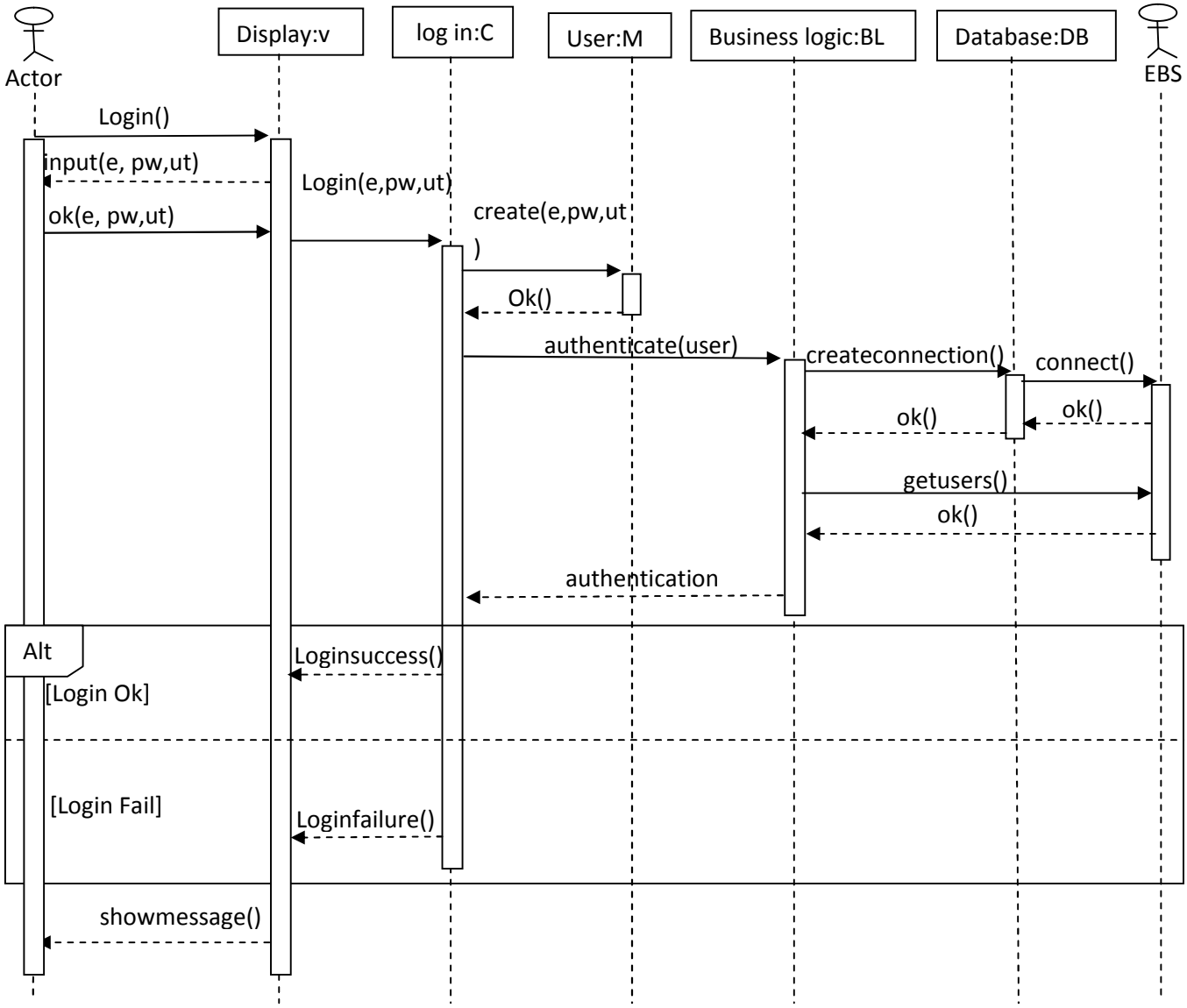


Figure 3 : Sequence diagram of Login

Here,e=email,pw=password,ut=user type

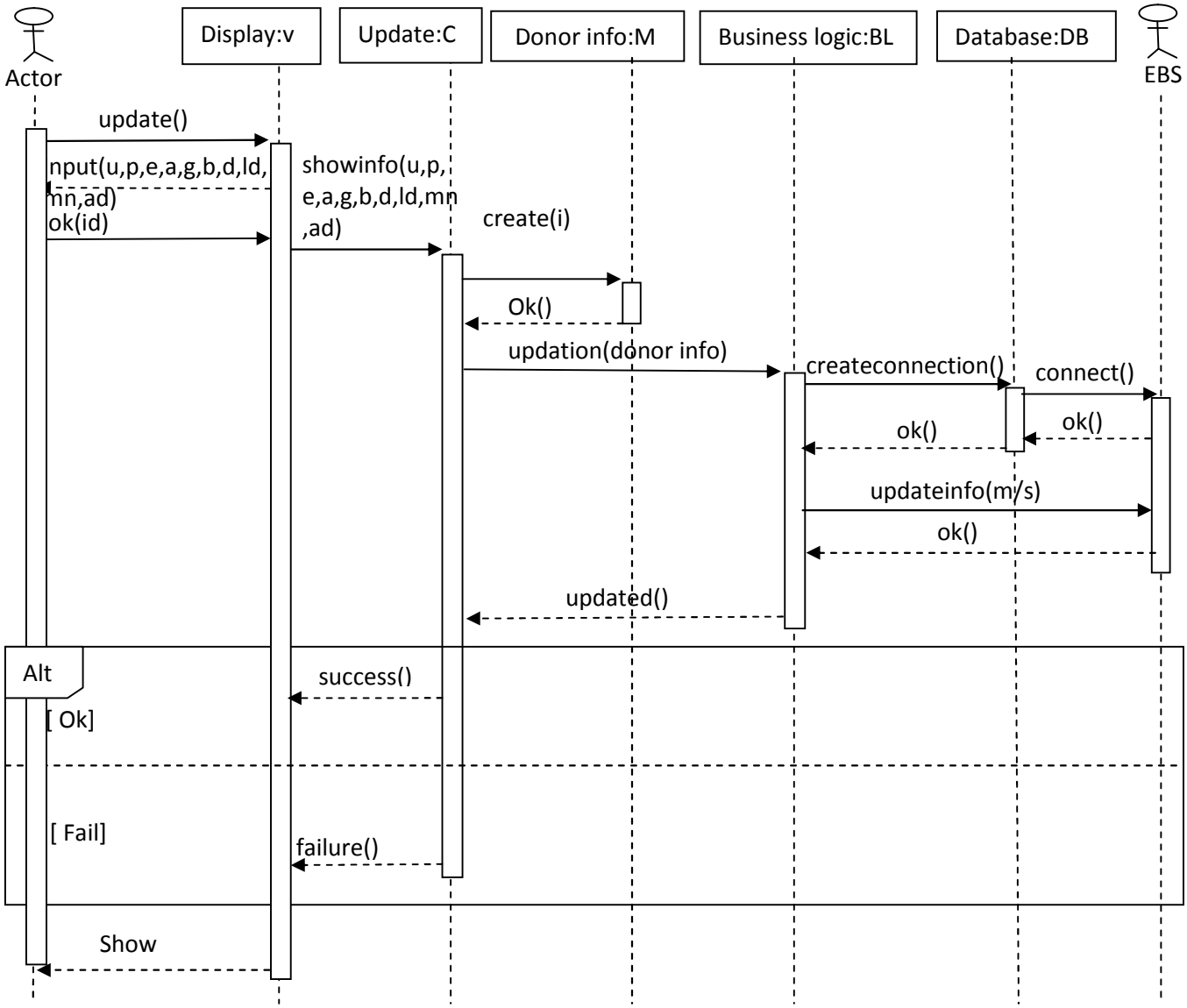


Figure 4:Sequence diagram of update donor info

Here,u=user name,p=password,e=Email,a=Age,g=Gender,b=blood group,d=district,ld=last donated date,mobile no,ad=address

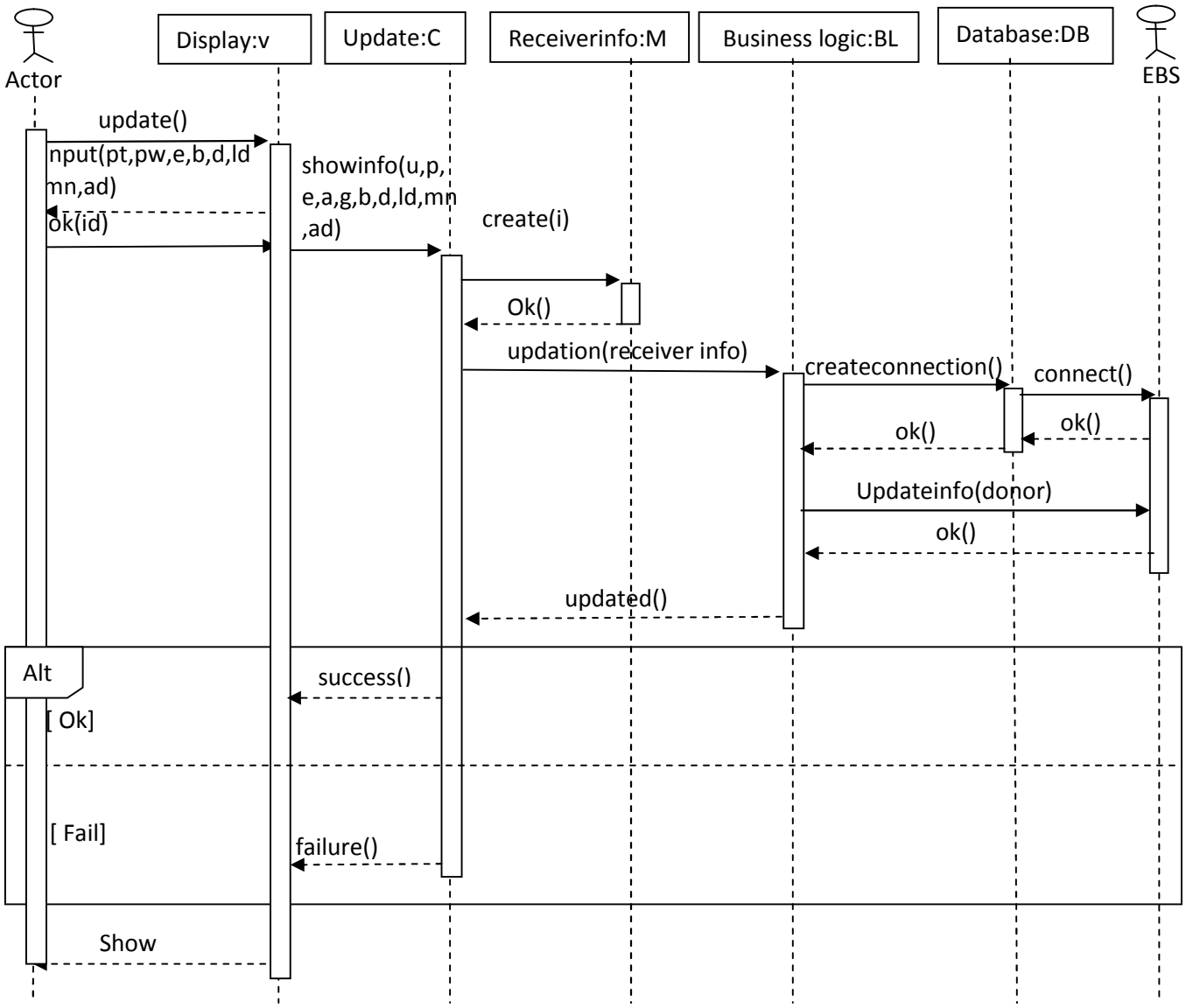


Figure 5:Sequence diagram of update donor info

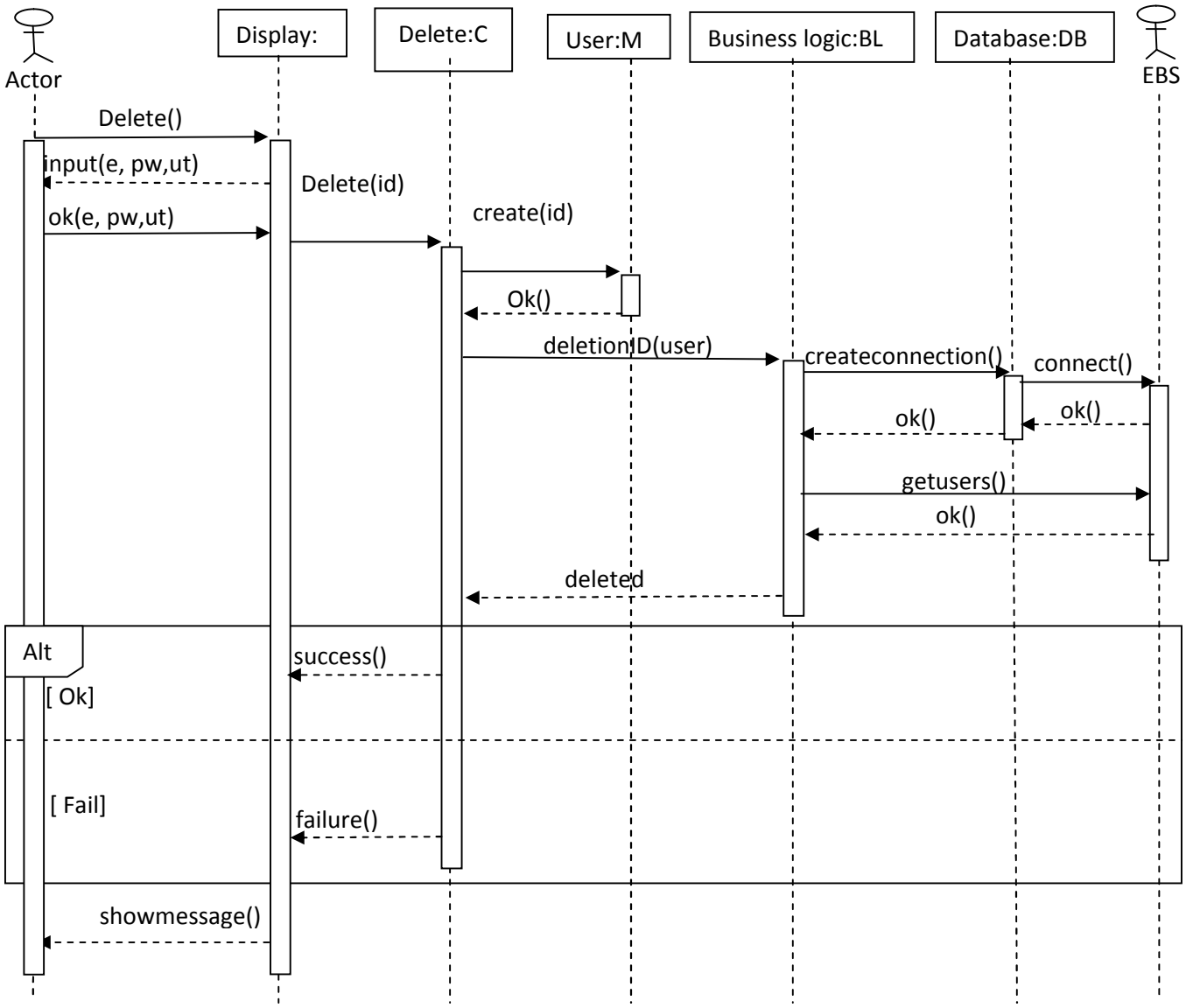


Figure 6:Sequence diagram of donor deletion

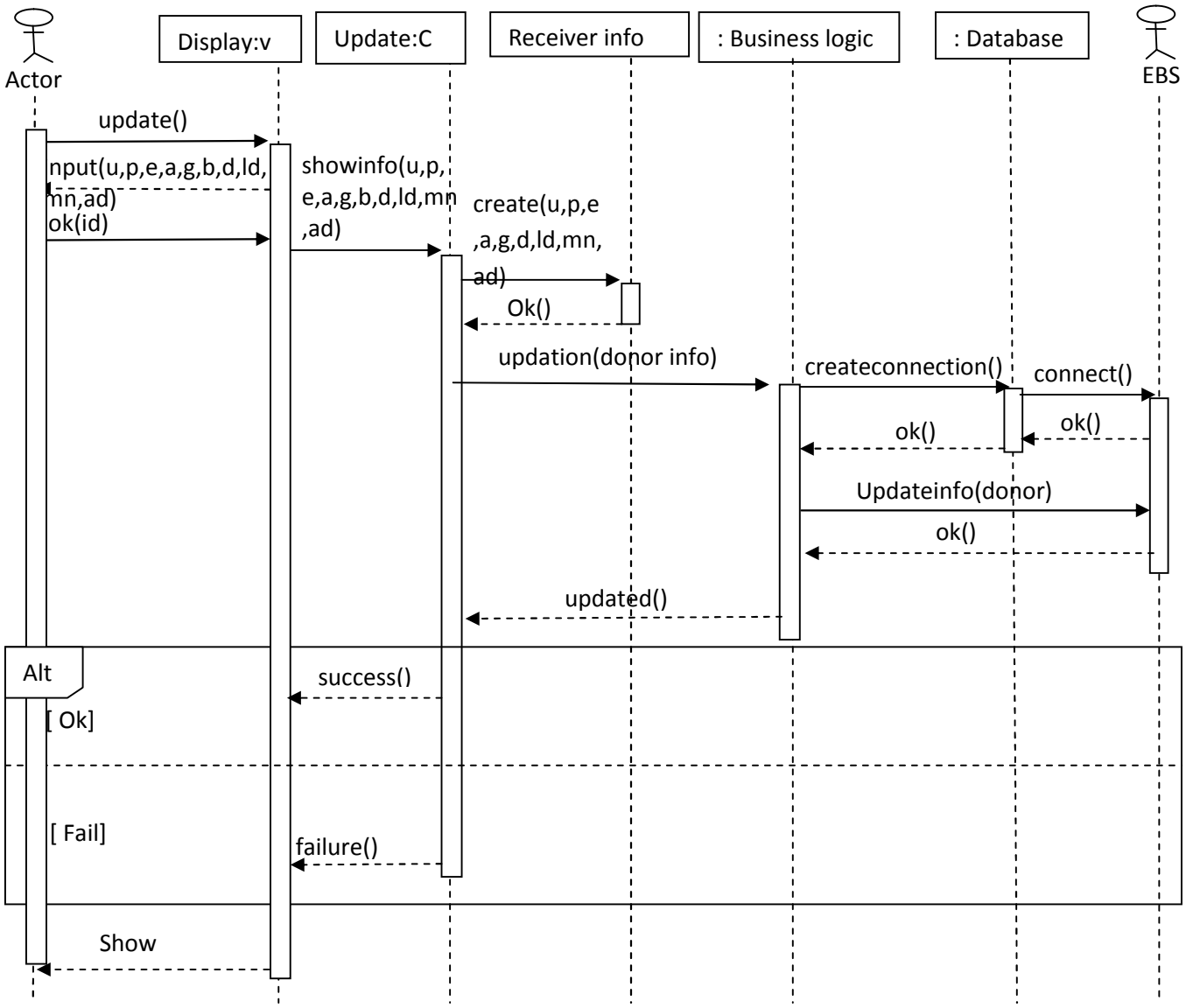


Figure 7:Sequence diagram of update receiver info

Here, Pt=patient name,pw=password,e=email,b=blood group,d=district,ld=last dated,mn=mobile no,ad=address

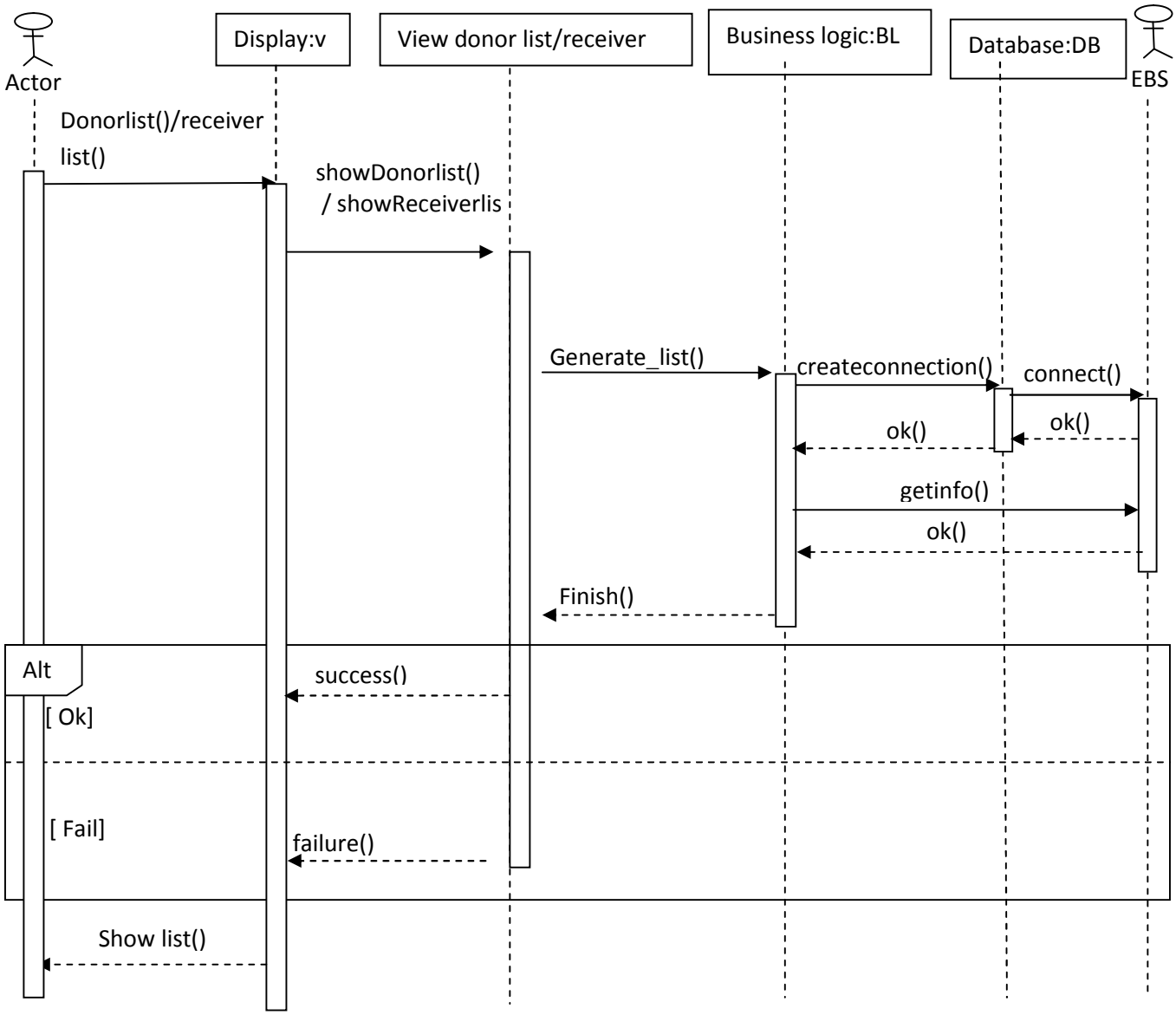


Figure 8:Sequence diagram of view donor /receiver list

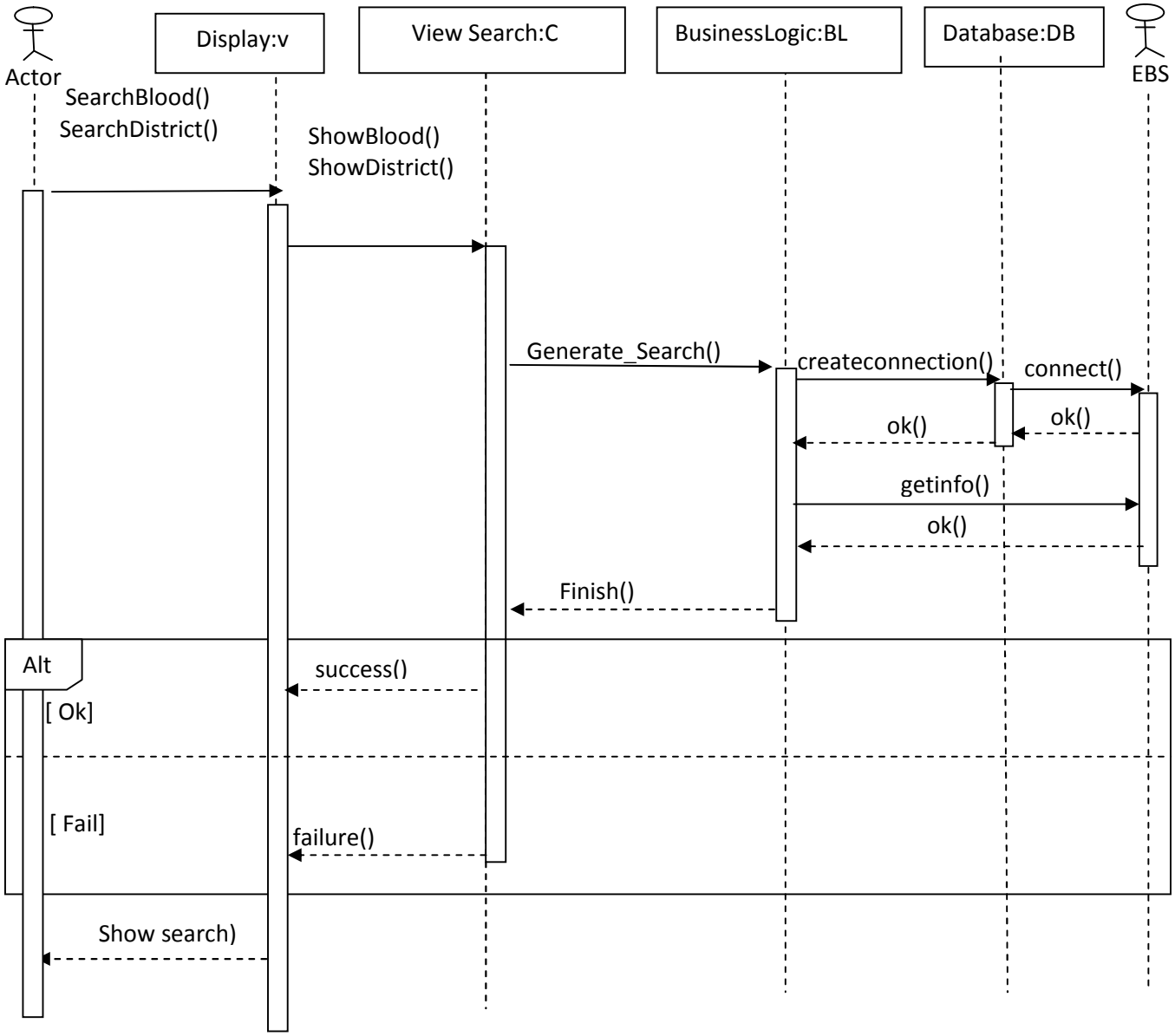


Figure 9:Sequence diagram of Searching Donor by blood/
Search Donor by District

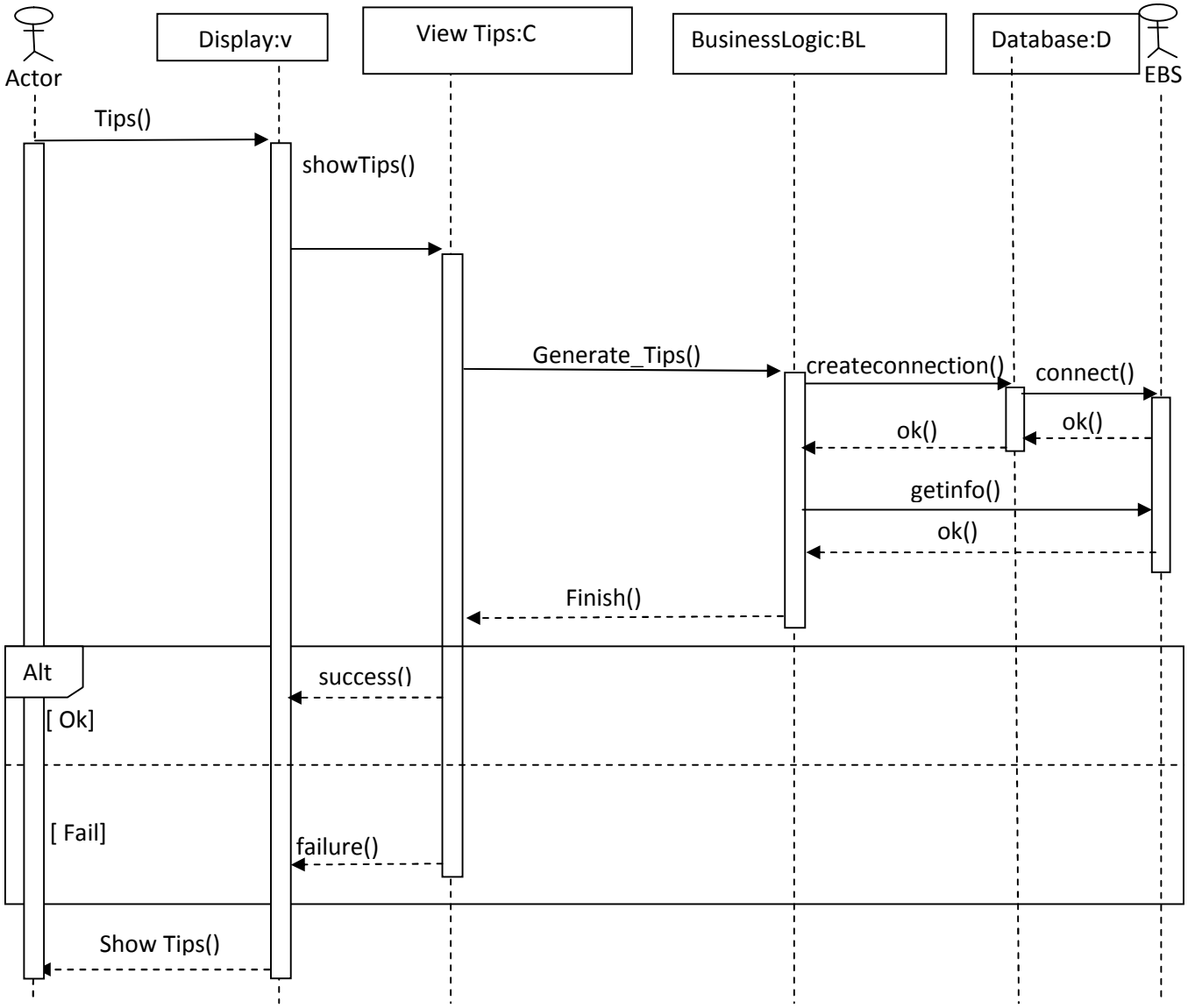


Figure 10:Sequence diagram of view Tips

3.3.2.Sequence Diagram for Admin Panel:

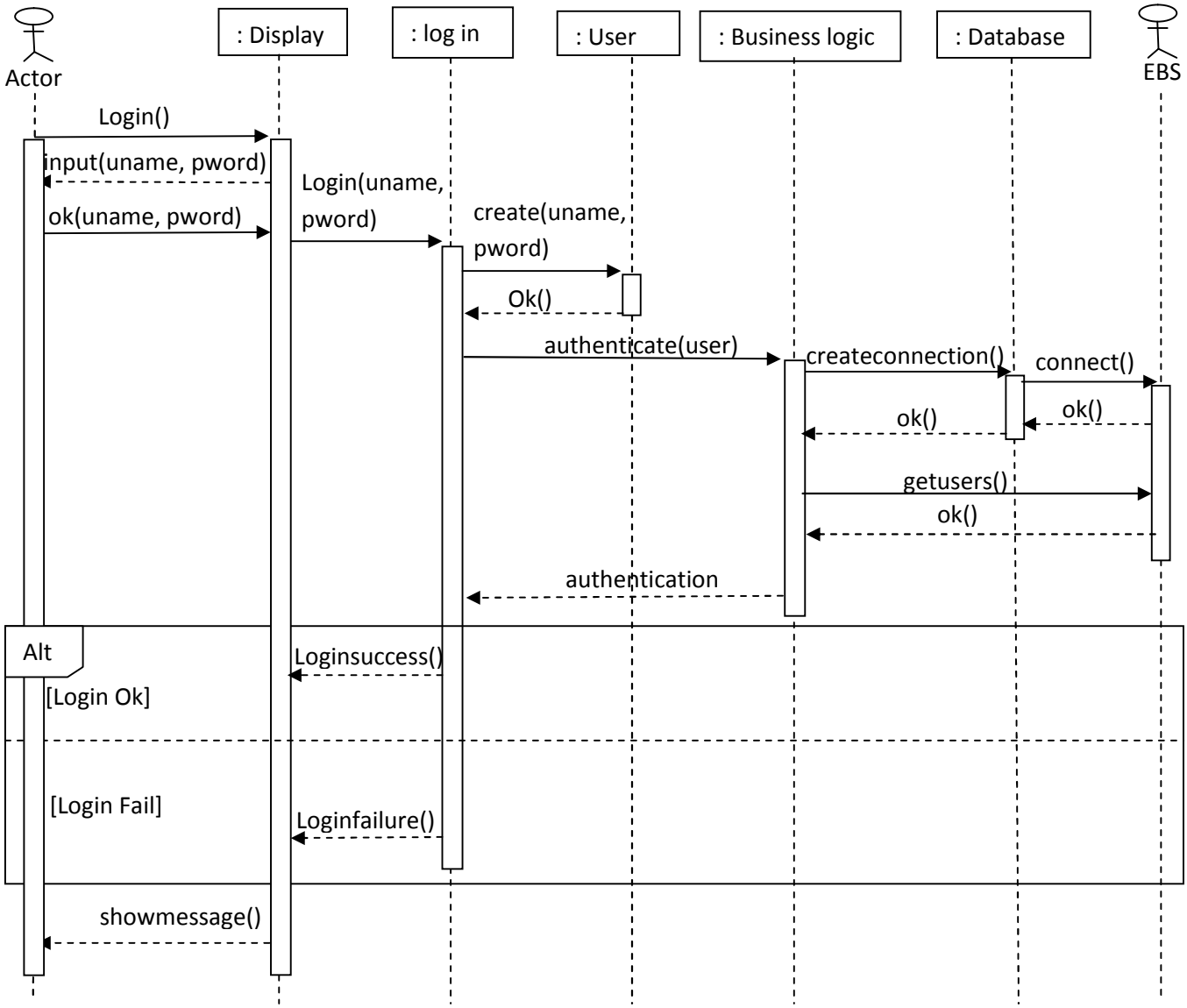


Figure 11:Sequence diagram of Login for Admin

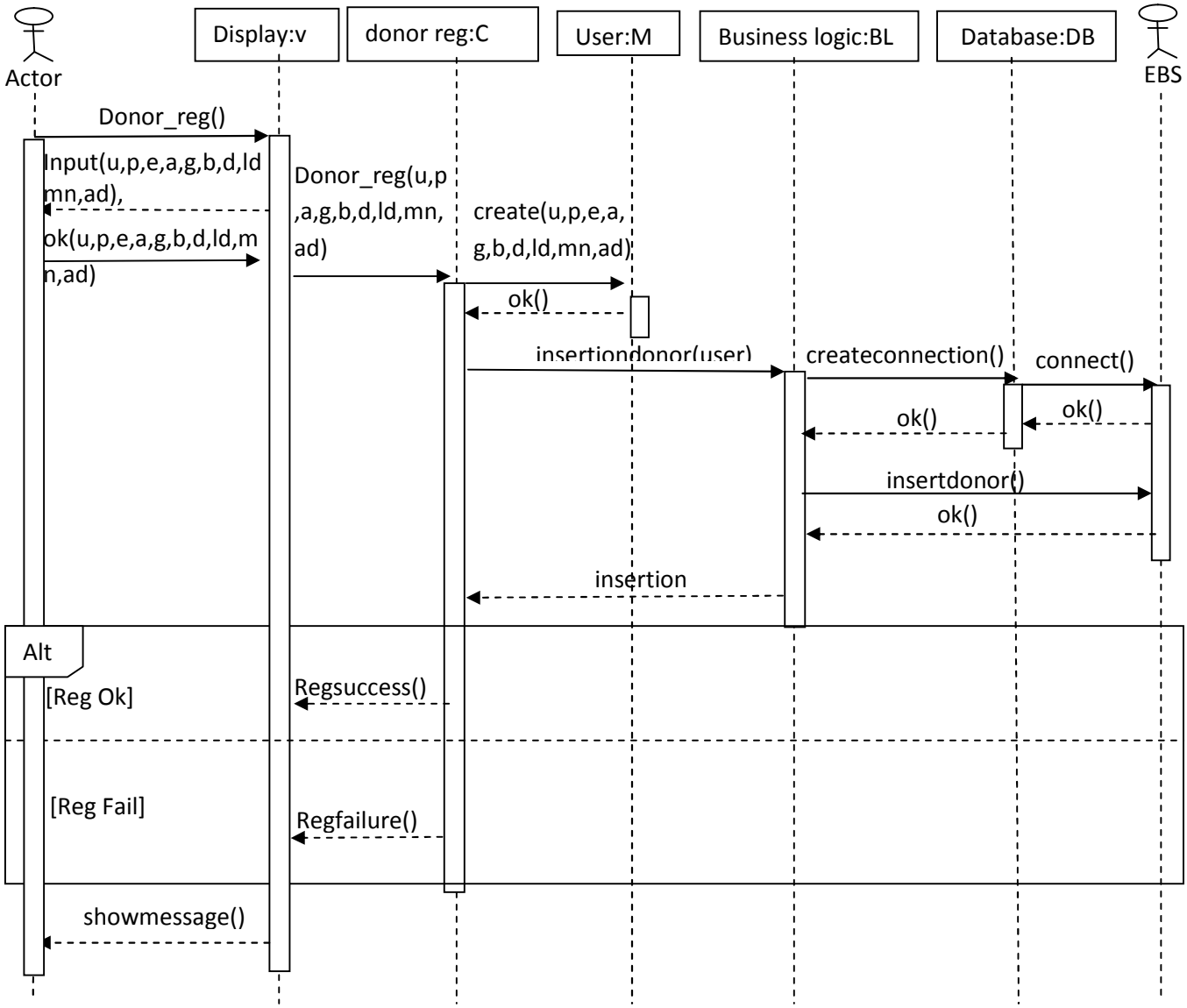


Figure 12:Sequence diagram of Donor registration For Admin

Here,u=user name,p=password,e=Email,a=Age,g=Gender,b=blood group,d=district,ld=last donated date,mobile no,ad=address

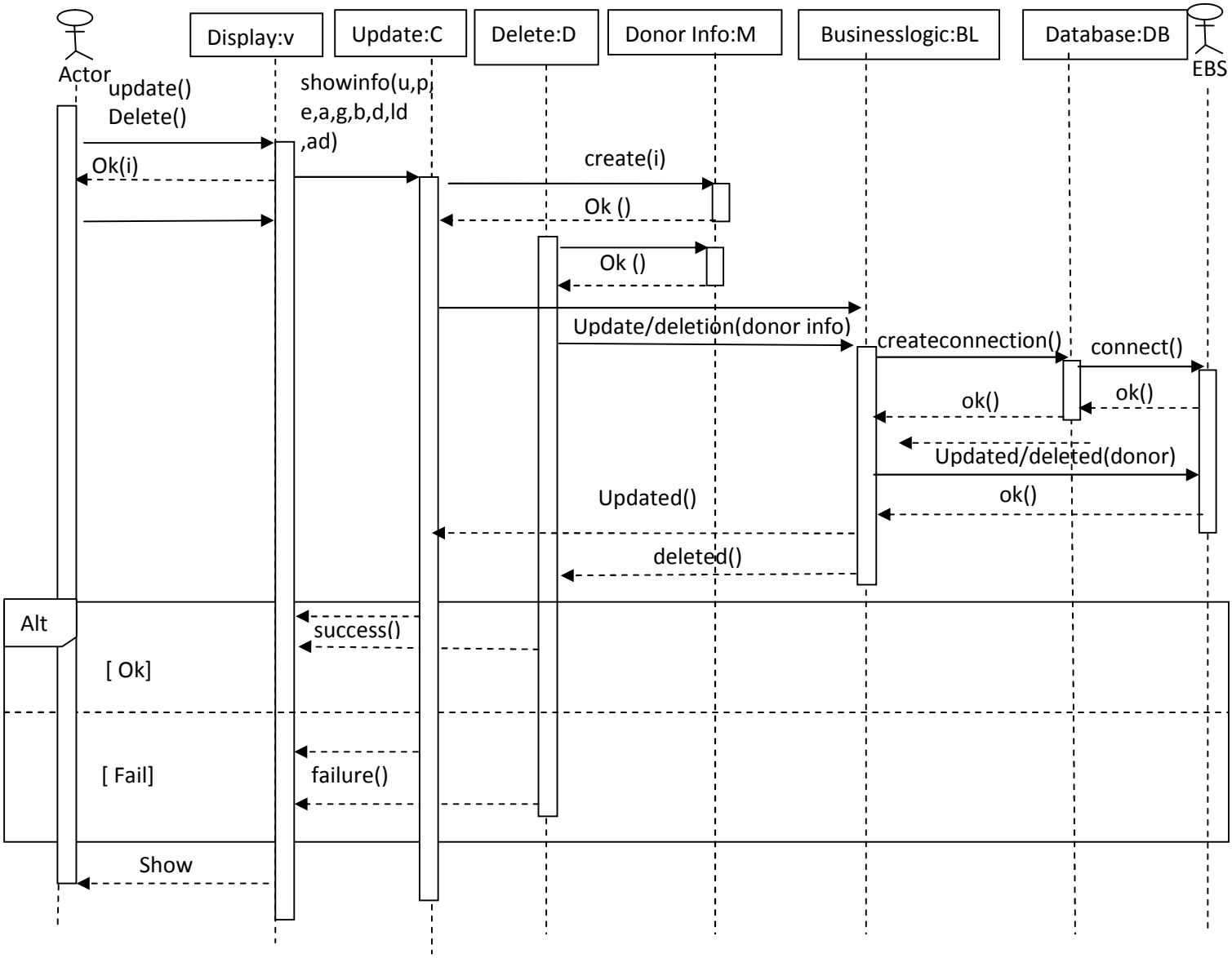


Figure 13:Sequence diagram of manage donor for Admin

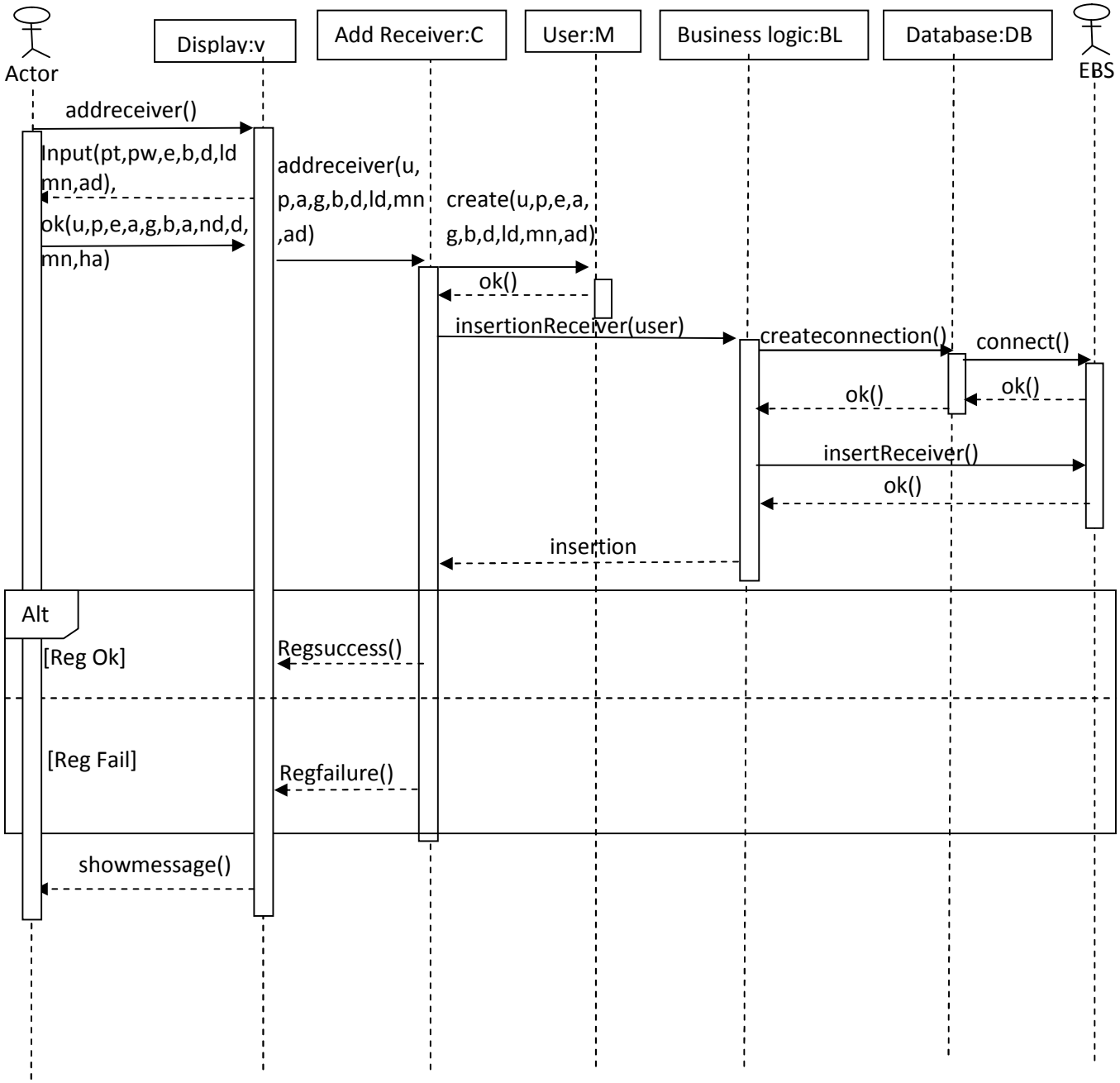


Figure 14:Sequence diagram of add receiver for admin.

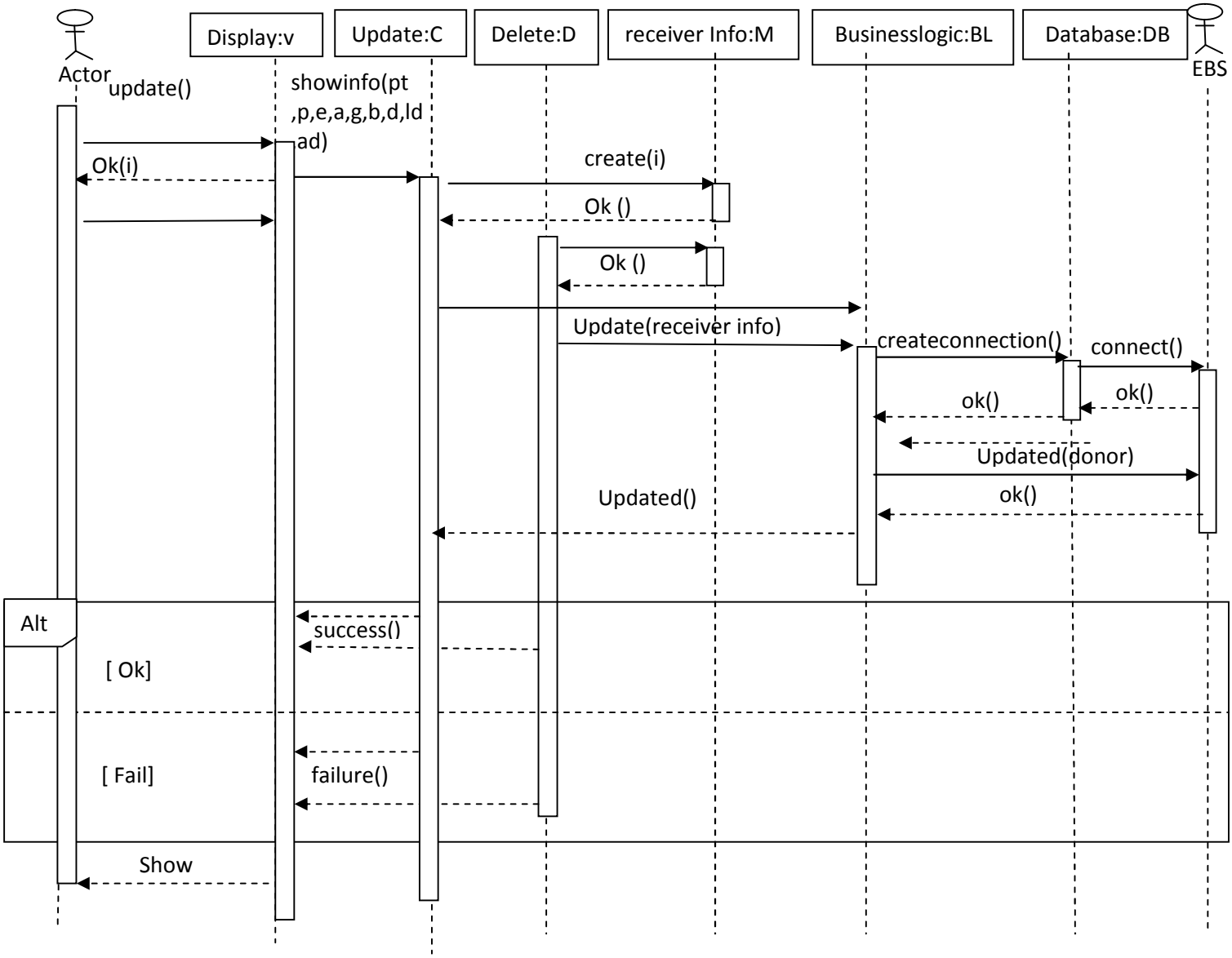


Figure 15:Sequence diagram of manage receiver for Admin

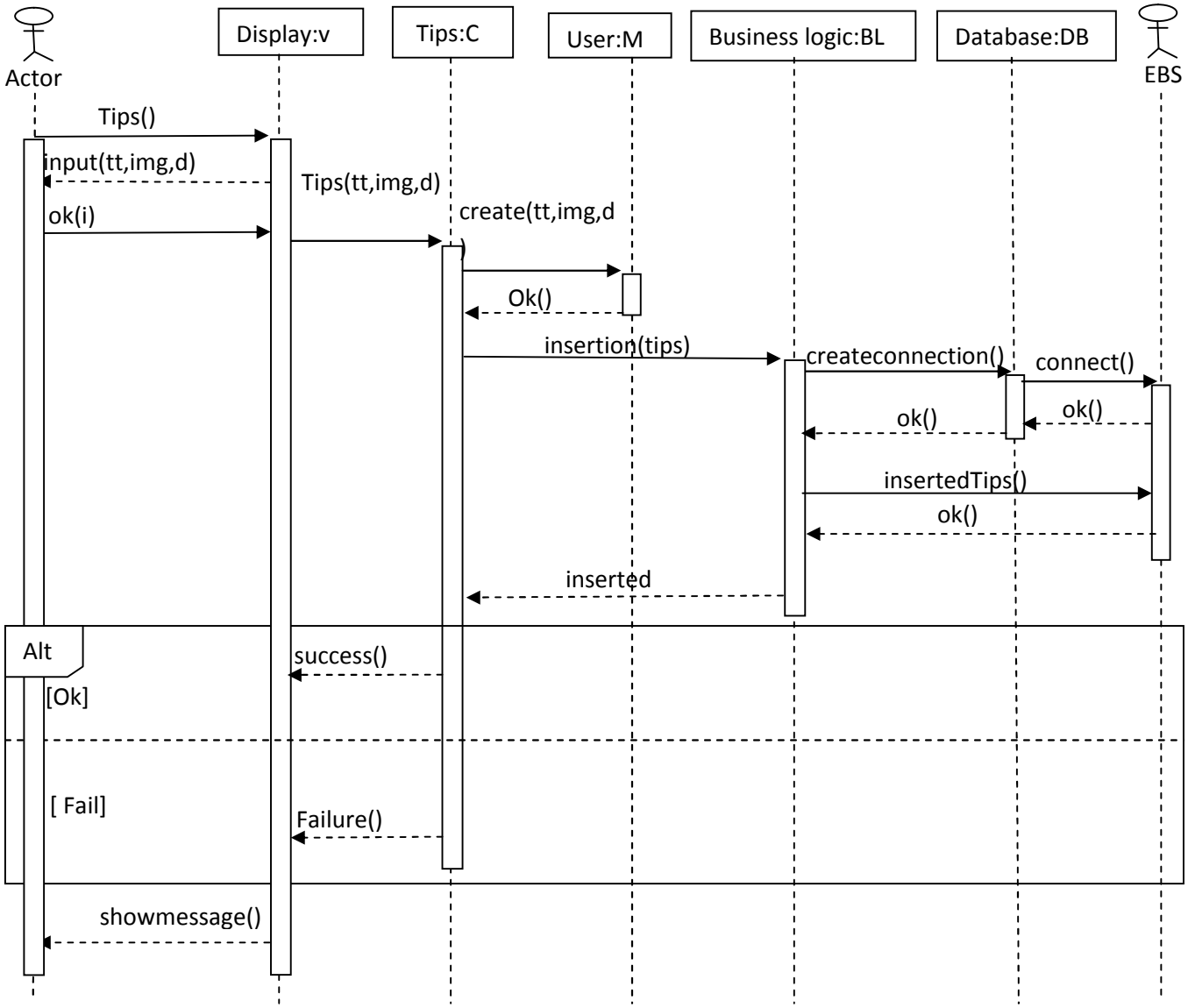


Figure 16:Sequence diagram of add tips for admin

Here,tt=Tips title,img=insert image ,d=description

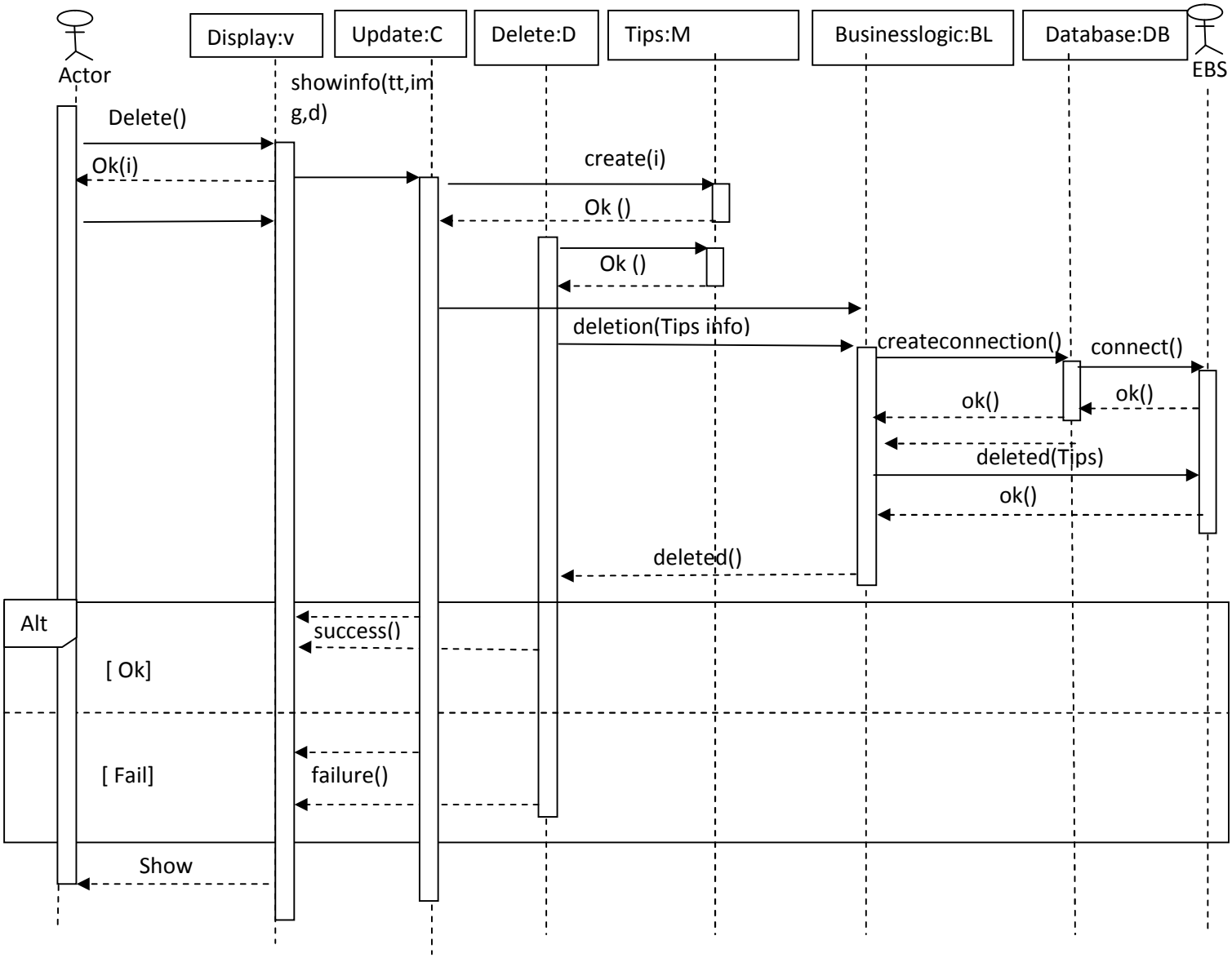


Figure 17:Sequence diagram of manage tips for admin

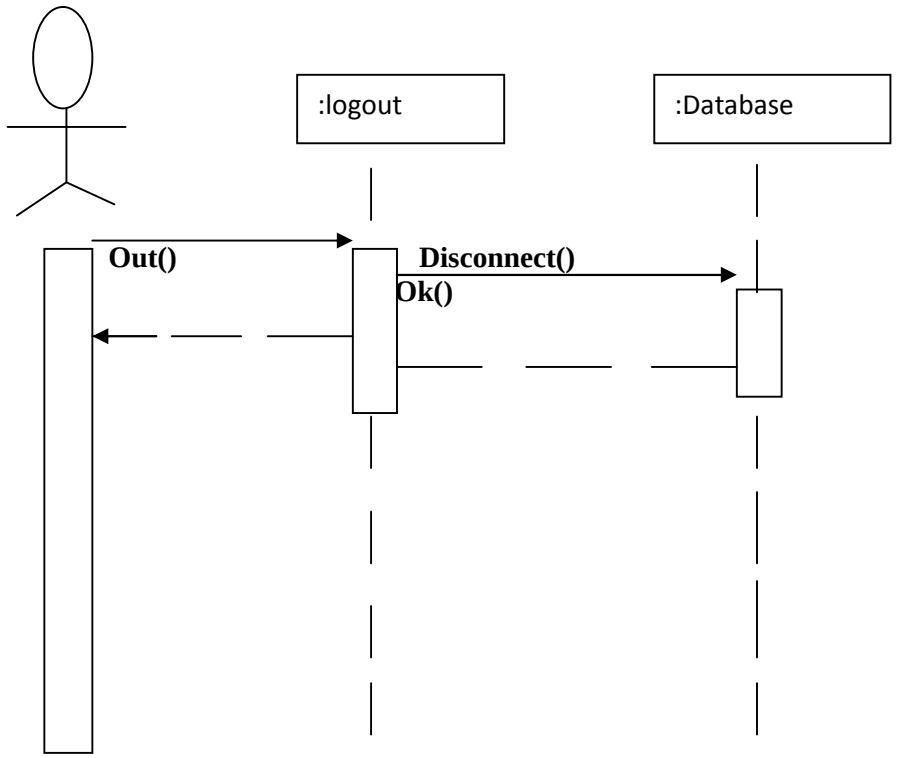


Figure 18:Sequence diagram of Logout

3.4.Architectural Pattern:

The e-blood System will be build by the MVC or Model View Control architectural pattern .Model View Controller or MVC as it is popularly called ,is a software architectural pattern
For developing web applications. A Model View Controller pattern is made up of the following three parts: the model, the view, the controller.

The model manages fundamental behaviors and data of the application. It respond to requests for information ,respond to instructions to change the state of its information, and even to notify observes in event-driven systems when information changes.

The view effectively provides the user interface element of the application.It'll render data from the model into a form that is suitable for the user interface.

The controller receives user input and makes calls to model objects an the view to perform appropriate actions.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented for e-Blood system which is given as follows.

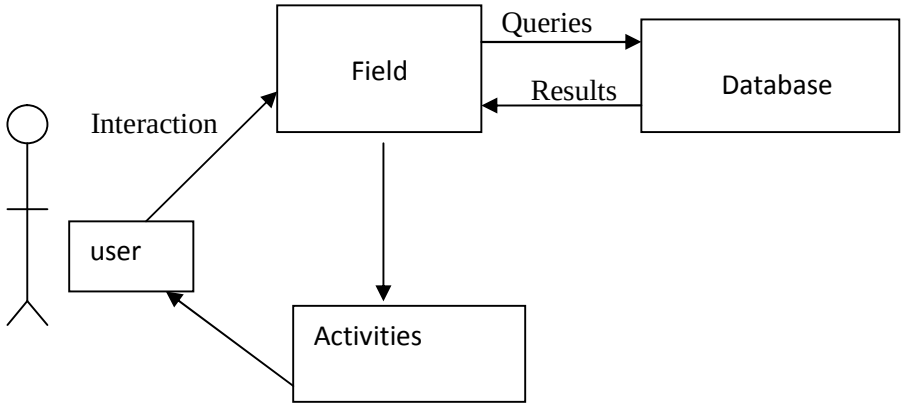


Figure : MVC Pattern in e-blood system.

3.5. Entity-Relationship Diagram:

An **entity-relationship (ER)** diagram is a specialized graphic that illustrates the interrelationships between entities in a database. It is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database and its requirements in a top-down fashion.

The ER diagram is principally used to capture the relationships that exist between static data objects in a problem domain or a design model. The basic elements involved in an ER diagram are defined as Entity, relationship, attribute. The entity-relationship(ER) diagram of e-

blood system are given below:

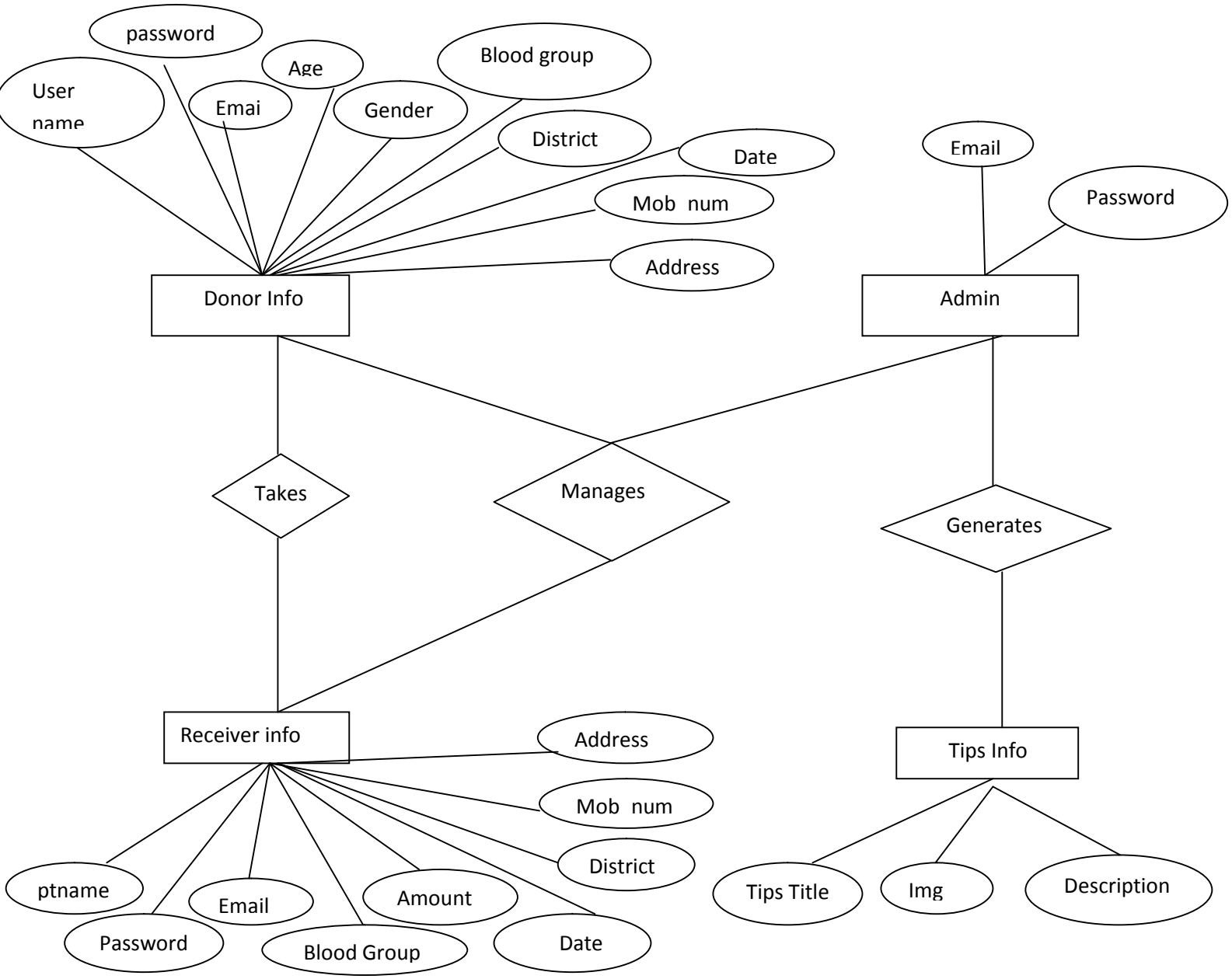


Figure: ER Diagram for e-Blood system

1. **Donor info:**User name ,password ,email, age, gender, blood group , district ,date, mobile number, address
2. **Receiver info:** patient name,password,email,blood group,amount, needed blood date , District, mobile number,Hospital address
3. **Admin:**email,password.

4. **Tips info:Tip**s title,img,description

5. **Descriptive Attributes:** email,needed date,age etc,

- **Relationship types**: Takes, manages, Generate are relationship type

3.6.Conclusion:

A detailed design document gives the overall structure of the system. Here we have included activity diagram to show the activity flow of the system, class diagram to show the classes of the system, sequence diagram to show the sequence of interactions of the system. This document will help us to overcome critical sections of the system and make the project efficient and practical.

CHAPTER4:IMPLEMENTATION

4.1.Introduction

Implementation is the important part of the software development. This can be done by implement our concept that include in the analysis and design state. All of this established as a program or a set of programs. To develop the software we select PHP as programming module and Html, CSS as scripting language for some several reason that are: Effective and efficient, Robust, Easily handle exception. We use MySQL for the database.

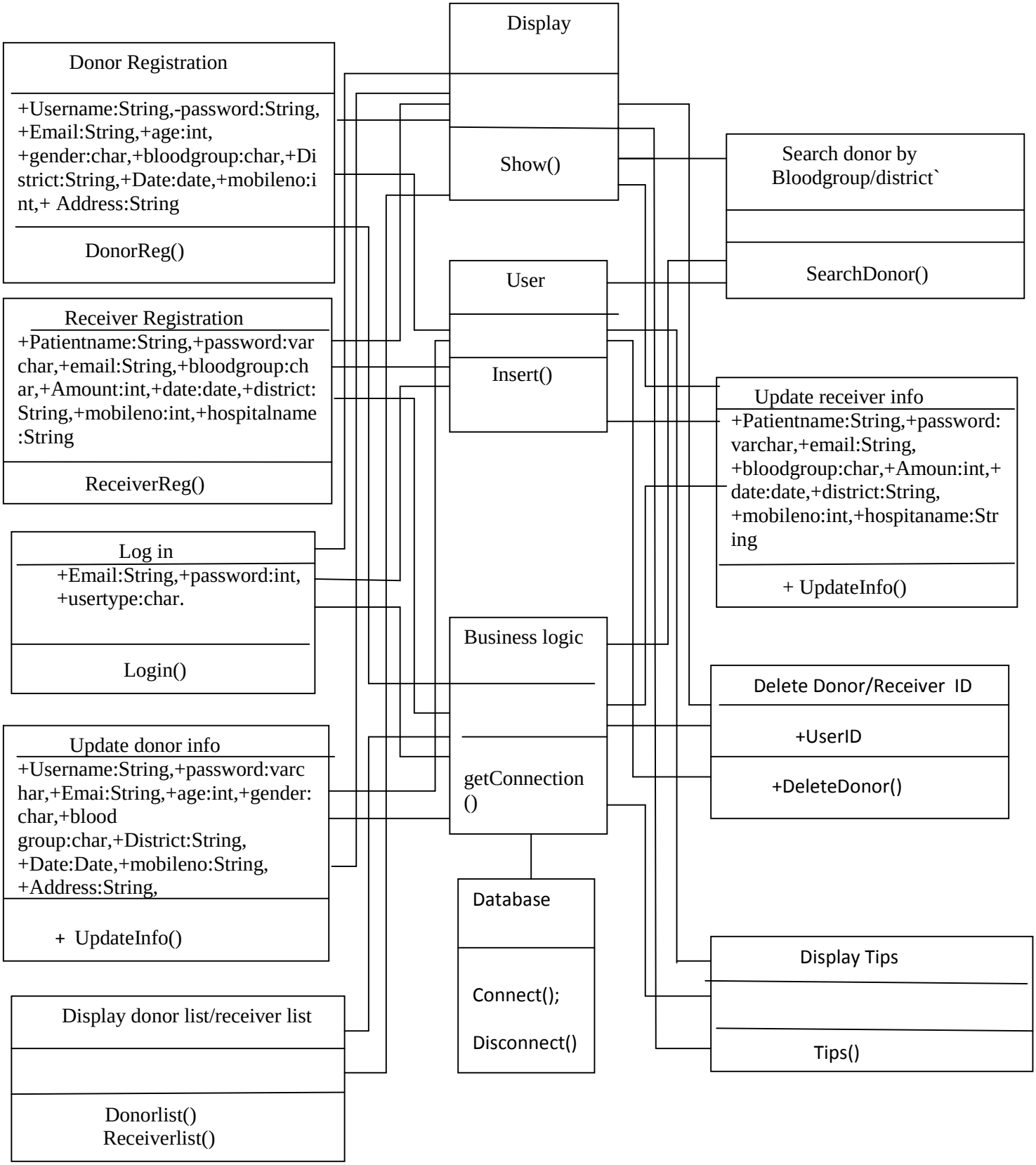
4.2.Detailed Class Diagram:

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system. With the share model facilities, you can reuse your class model in the interaction diagram for modeling the detailed design of the dynamic behavior. The Form Diagram allows you to generate diagram automatically with user-defined scope[6].The Detailed class diagram for our system are provided by three module: Admin,Donor ,Receiver/Patient.

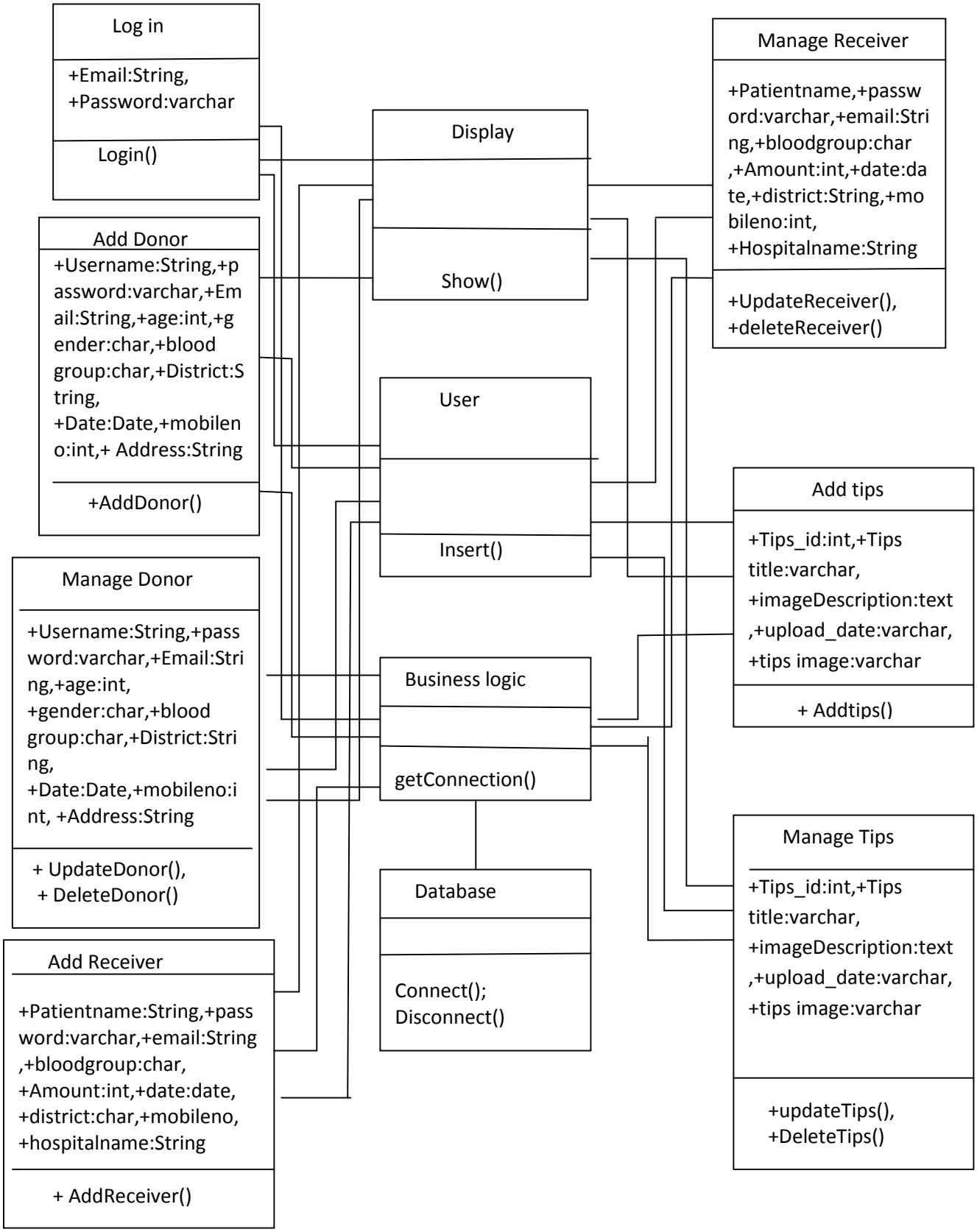
The Admin module has details of the donors and the receiver/patient. The donor and the recipients are provided with the particular identity and password. The member will be provided a login interface for the entry of the Id and password. The functions are there in the module like Change password, manage donor details, manage referrals, update and upgrade donor details, view the reviews, logout.

The Donor and receiver module also provides the Id and the password which is given to every particular registered member.Every user can view donor and receiver list, can search donor by bloodor district. User can know about donating blood,heath tips. By login to the module the user can perform for each profile functions like change password, mobile number, address etc.

4.2.1.Detailed Class Diagram For Donor & Receivers:



4.2.2.Detailed Class Diagram For Admin Panel:



4.3.Used Software Process Model:

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective as: Specification, design, validation and evolution. [6]

The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use. There are various Software development models or methodologies

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or overlapping steps.

4.4.Used Design Pattern:

Design patterns represent the best practices used by experienced object-oriented software developers. Design patterns are solutions to general problems that software developers faced during software development[7]..For this we have decide to use singleton pattern in our system. The Singleton pattern is probably the most famous and at the same time the most controversial pattern known to us. It must be also be the simplest pattern to learn and implement. Like any other pattern, Singleton exists to solve a common business problem that is ‘managing the state of a resource’.

One of the toughest issues to debug is the one created by the multiple instances of a class which manages the state of a single resource. It is highly desirable if we can use some Design Pattern to control the access to that shared resource. The Singleton pattern fits the bill perfectly to solve this scenario; by wrapping a singleton class around this problem ensures that there will be only one instance of the class at any given time. A most common and clichéd example for a singleton

class is the one used for logging purposes where the whole application needs only one logger instance at anytime.

4.5.Demonstration of the System:

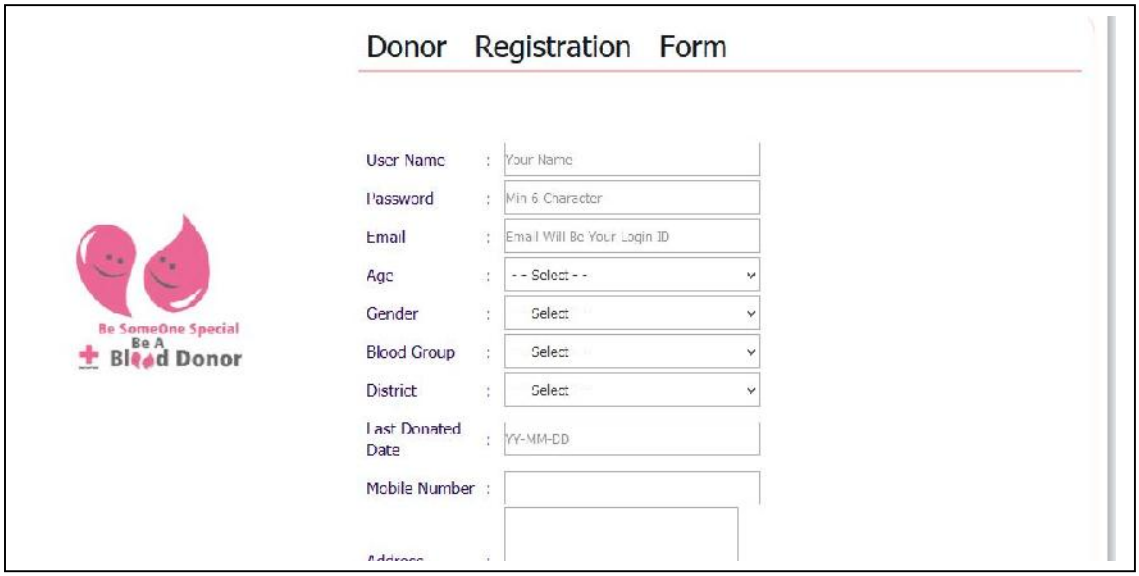
This chapter gives an overview of the e-blood system software through snapshots of user interfaces and form operations descriptions .On the first section the snapshot will given for donor and receiver

Home Page:

This page is main interface of the software which detects user which purposes are possible through the software. This form contains buttons, Icons and Menu Bar. By Clicking respective button user can access required form.

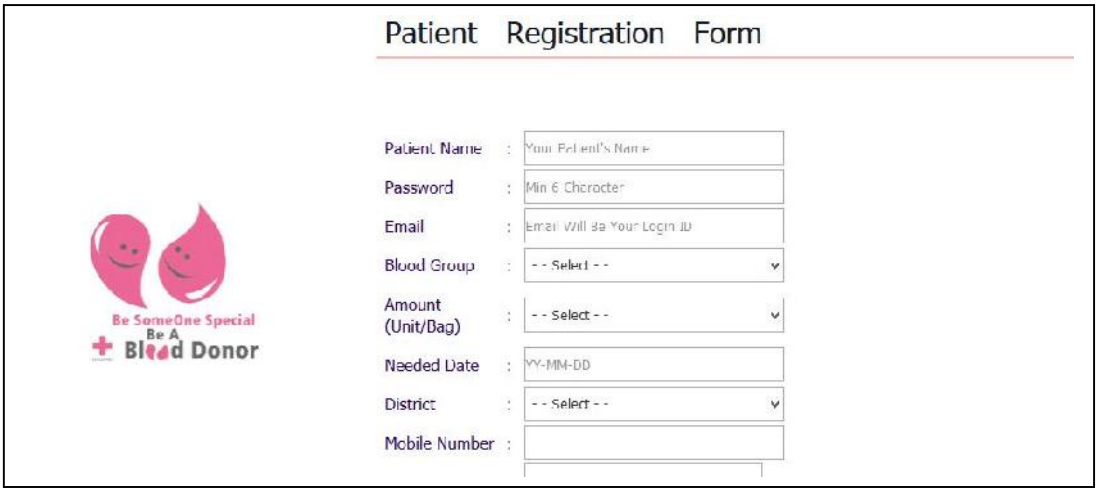
Donor Registration Form:

Description:this page show donor registration form for the user**.**



Patient Registration Form:

Description:this page show receiver/patient registration form for the user**.**

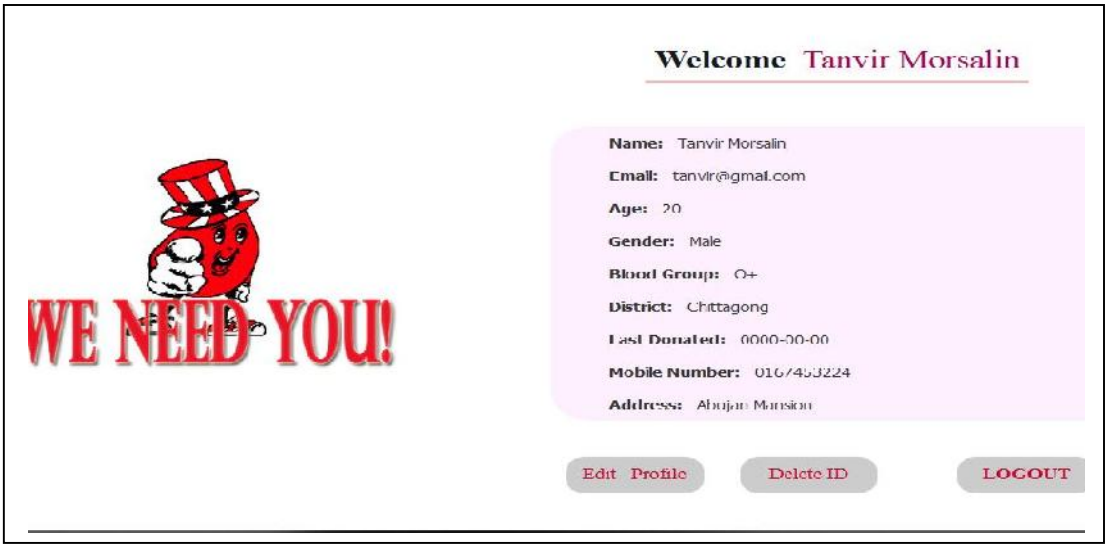


Login:

Description: This page show donor and receiver/patient login interface**..**

Donor Profile:

Description: When donor login on his/her profile ,this interface will show on the system

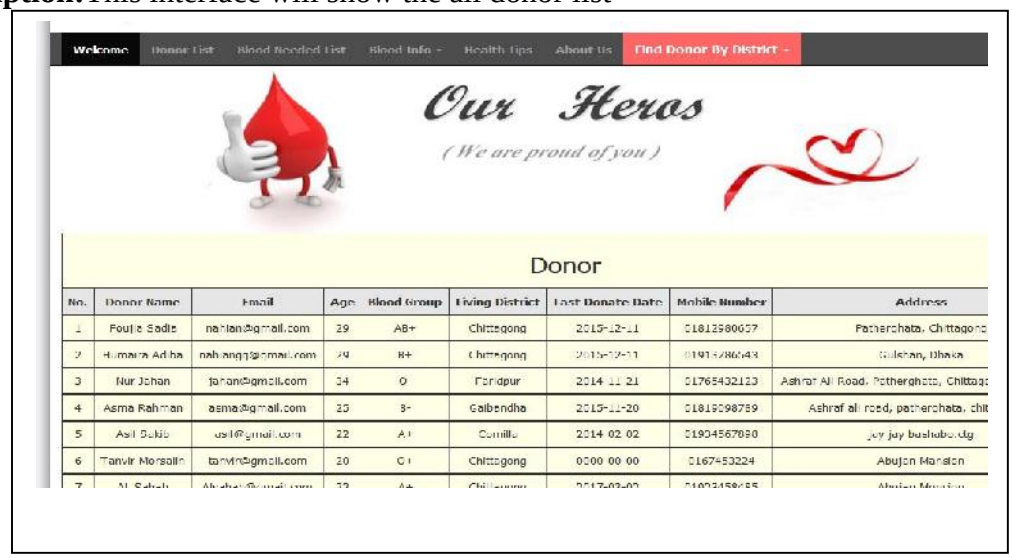


Donor Update Form:

Description: When donor need to update, he can update by accessing on his profile update interface

View for Donor List:

Description: This interface will show the all donor list

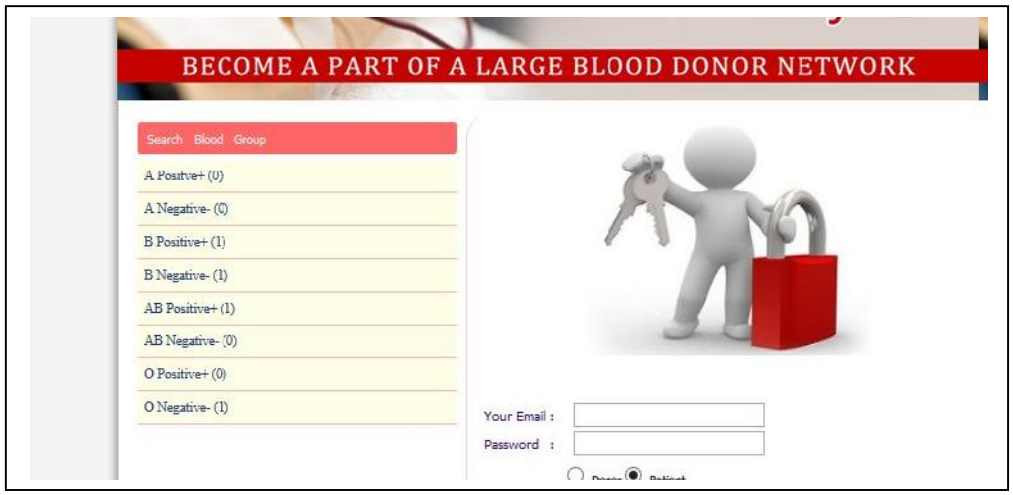


View for Receiver/Patient List:

Description: This interface Show all receiver/patient list

Search By Blood Group:

Description: On the home page user can search for blood according to blood group .This facility also given on the menubar.

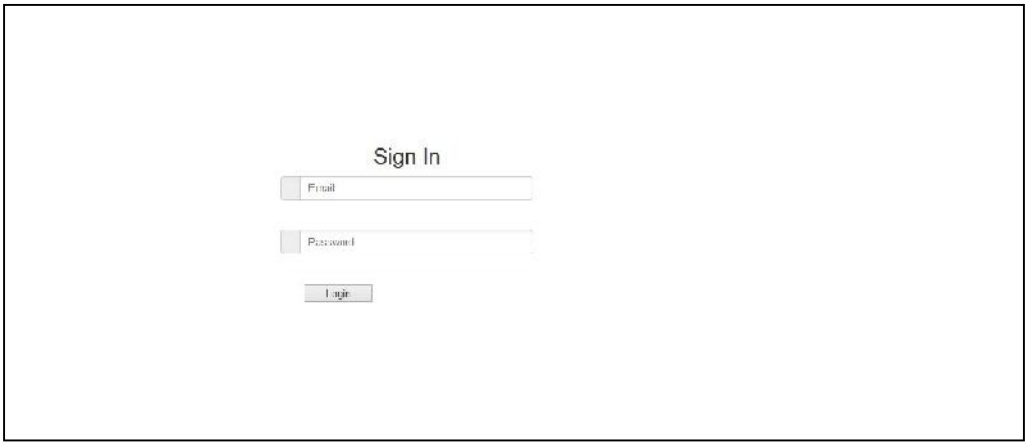


Search By District:

Description: On the home page user can search for blood according to district.This facility provide on menubar

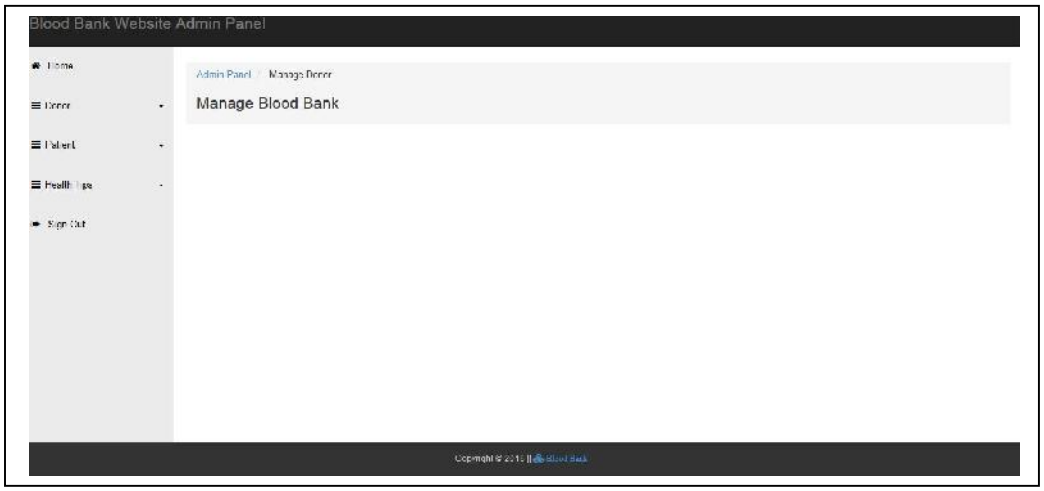
Admin Panel Login:

Description:this is for admin panel login interface. By providing admin email and password he/she can access his/her admin panel



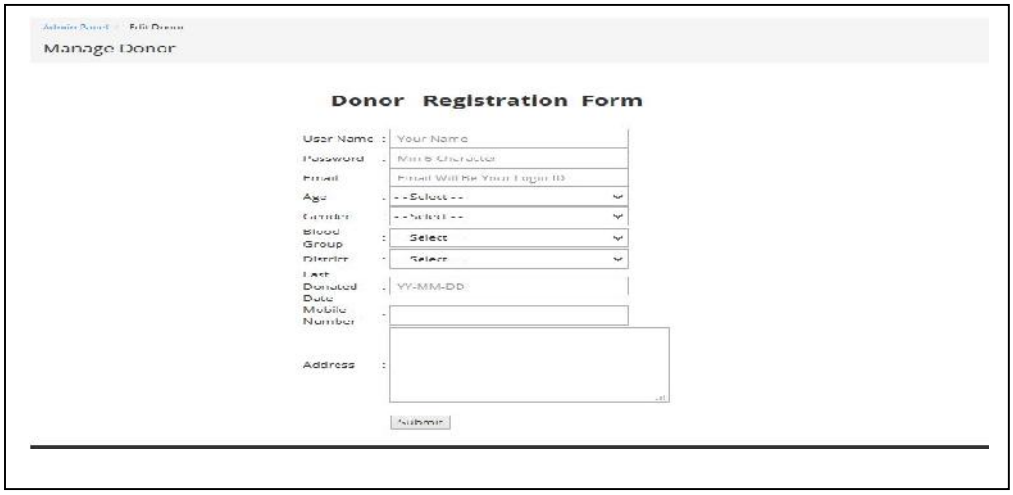
Admin Home Page:

Description:This interface called admin panel.From here he/she can control his /her website.



Add Donor:

Description:This interface provide donor registration form for the admin

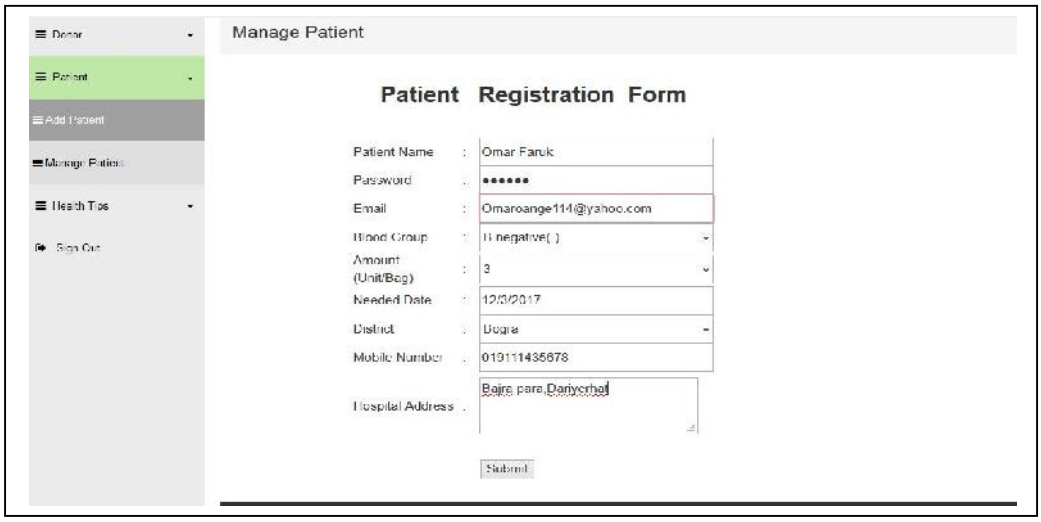


Manage Donor:

Description:This interface provide update or deleting donor case for the admin .According to his/her deman he/she can update/ delete donor information.

Add Patient/Receiver:

Description: This interface provide receiver/patient registration form for the admin

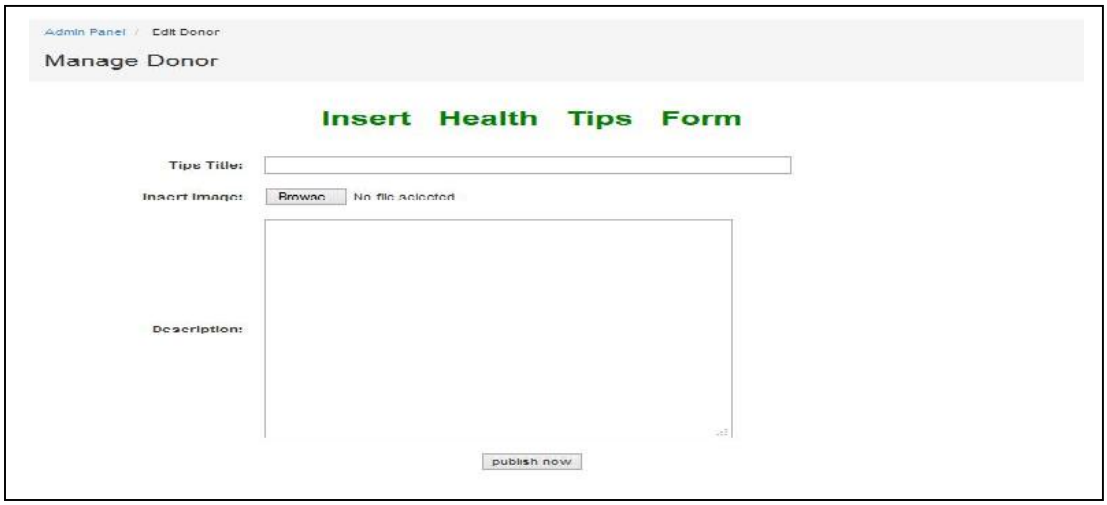


Manage Patient/Receiver:

Description: This interface provide update or deleting patient/receiver case for the admin .According to his/her deman he/she can update/ delete receiver information.

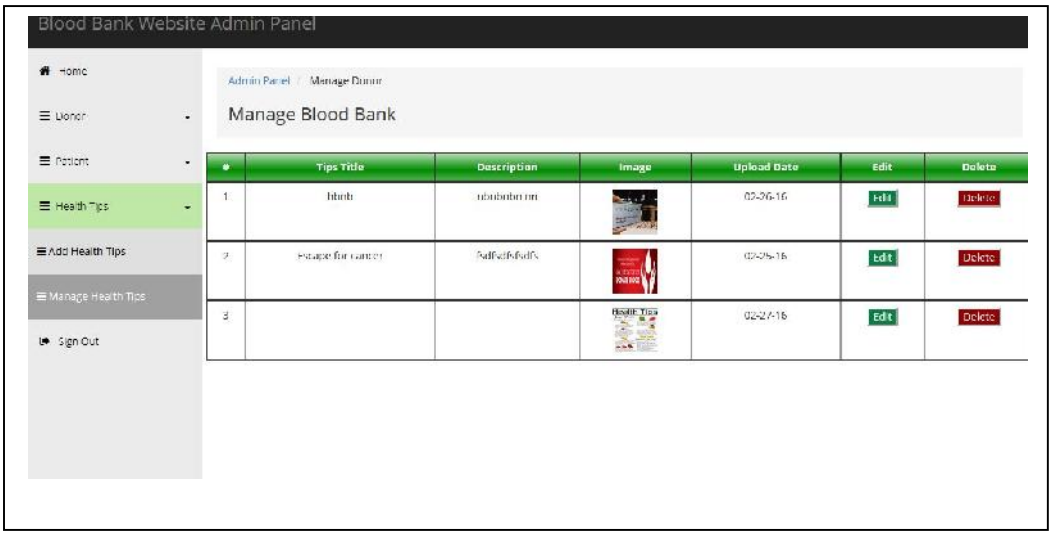
Add Heath Tips:

Description: admin add health tips by using this interface.



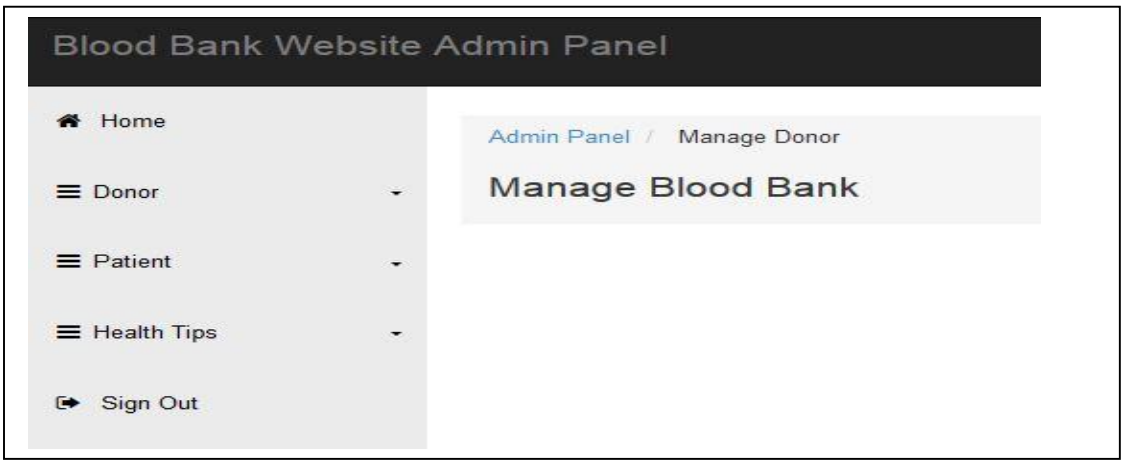
Manage Health Tips:

Description: This interface provide update or deleting health tips case for the admin .According to his/her necessary he/she can update/ delete heath tips information.



Log Out:

Description:On clicking sign out admin can come out form admin panel.



4.6. Testing:

Testing is the process, which is used to uncover errors from software that are made inadvertently as it was designed and constructed. There are two basic type of testing. These are:1) Static testing,2) Dynamic testing

Static testing is related with inspecting the code. For this software, all the codes are reviewed several times. There are two types of dynamic testing. These are :a) White box testing

b) Black box testing

White box testing:

Software is tested from two different perspectives:

1. Internal program logic is exercised using “white box” test case design techniques.
2. Software requirement are exercised using “black box” testing design techniques.

In both cases, the intent is to find be maximum number of errors with the minimum amount of effort and time. White box testing of software is predicated on close examination of procedural detail. Logical paths through the software are tested by providing test cases that exercise specific sets of condition and /or loops. The status of the program may be examined at various points to determine if the expected or asserted status corresponds to the actual status.

White Box Testing Techniques:

White box testing sometimes called glass box testing is a test case design method that uses the control structure of the procedural design to derive test cases.

Using white box testing methods, software engineer can derive test case that:

1) Guarantee that all independent paths within a module have been exercised at least one.
2) Exercise all logical decisions on their true and false sides
3) Execute all loops at their boundaries and within their operational bounds, and
4) Exercise internal data structures to ensure their validity.

Used Following method:

In our project we follow white box testing. We have knowledge of the source code; it becomes easy to find out which type of data can help in testing the application effectively .It helps in optimizing the code. Extra line of code can be removed which can bring in hidden defects.Due to my knowledge about the code, maximum coverage is attained during test scenario writing.

Test Log:

Name: user Login				
NO.	Test condition	Expected Result	Actual Output	Status (pass.fail)
Test 1	Click on submit button without user name and password	System does not allow user to login	System displays message and resume to the same page	Pass
Test 2	Click on submit button with invalid user name or password	Message “pleas e fill up the user name and password”	As expected	Pass
Test 3	Click on submit button with correct user name add password	System allow user to login	System allow user to access application based on right given to him	Pass

Name: Registration				
NO.	Test condition	Expected Result	Actual Output	Status (pass ,fail)
Test 1	Click on submit button without user name and password	System does not allow user to login	System displays message to the same page	Pass
Test 2	Click on submit button without correct password & re password	Message “please fill up the correct password and re password”	System displays message and resume to the same page	Pass

Test 3	Select on user type with correct user registration	System allow user to login	System allow user to access application based on right given to him	Pass

Name: User Activities				
NO.	Test condition	Expected Result	Actual Output	Status (Pass,fail)
Test 1	Click on update button without add new information of user	System does not allow to save data without add update information	System displays message and resume to the same page	Pass
Test 2	Click on delete button	Message “Delete successfully”	System allow user to add more information into the system	Pass

Name: Receiver/Patient Request				
NO.	Test condition	Expected Result	Actual Output	Status (Pass,fail)
Test 1	Click on update button without add new information of request for blood	System does not allow to save data without add update information	System displays message and resume to the same page	Pass
Test 2	Click on donor list	System display donor list	System allow user to show donor list	Pass

Name: Donor Request				
NO.	Test condition	Expected Result	Actual Output	Status (Pass,fail)
Test 1	Click on update button without add new information of donor request for blood bank	System does not allow to save data without add update information	System displays message to the same page	Pass
Test 2	Click on blood needed list	System display Blood needed list	System allow user to show blood needed list and donor can willingly contact with receiver	Pass

System Requirements:

Any person who wants to run e-blood system in his/her computer then he/she must have the following minimum software and hardware requirements.

Hardware Requirements:

** Processor: 1 GHz
** RAM: 1 GB
** Hard Disk: 20 GB

Software Requirements:

Editor:notepad++
Web Design:HTML,CSS
Scripting Language:Javascript,ASP,PHP
Server:WAMP server

4.7.Conclusion :

The implementation processes also contains software preparation and transition activities, such as the conception and creation of the maintenance plan also, the preparation for handling problems identified during development, and the follow- up on product configuration management.

CHAPTER5:CONCLUSION

It has been a great pleasure for us to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in and PHP web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with **"Blood Bank Management System".** It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

5.1.Results and Discussions

Various reports were generated through this prototype software. The reports were generated both in tabular as well as in graphical form. This system generates various reports as: . Registered Donors , Blood Donated Report , Donors' Confirmation Report, Donors Recipient Report, Areas-wise and blood groups-wise Report , Gender-wise Analysis, Professional-wise Analysis and City-wise Analysis. The donor record view displays details about the member donors. This result is useful both for the administrators as well as patients to know the donors contact numbers, as well as his group. Computers are gradually becoming accepted in about every field of life, like business environment, health, industries, and research environment. Within these application areas, online systems appear to have become most suitable for health care and life saving processes. The proposed system is able to manipulate these real facts. These analyses reports show that most of the peoples are unaware of the utility of such activities. In this context seminars and wide publicity is required both in male and female.

5.2.LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

5.3.Future Work:

There is always a scope of betterment and the candidate system is not against this perception. This project is especially designed for Blood Bank Management to take

appropriate step to improve working standard and documentation through computerization. Since our system is going to be implemented as the application of “Blood Bank Management System” on PHP as frontend and MySql as backend tool, so it will help better for both user as well as developer .As per as my project cover all important details of Donor information, Stock, and Delivery; therefore there is some future improvement area exist also.The main scope of future may be:

- Adding Screening module in my software which help to test the collected blood.
- Fully automated, need for human intervention
- Reducing manual efforts for maintain the system
- More user friendly and fixing unconscious bug.

5.4.Conclusion:

We have tried at best as possible to give them a best solution with my capability, industry and patience. In this solution, some limitations may have. But We have tried to overcome the limitations as per as possible. Quires are efficiently done and my project shows fast query. More over our software is user friendly. So we can claim that my software project shows better performance. In e-blood system, we tried to hard to include all the components which actually need in a “Blood Bank Management System”. This is our first project and we are inexperience so we may have mistake or have some problem in the software. We think and hope that we will develop more perfectly in the next version when we get the user feedback.

References

[1] http/www.donateblood.com.au/
[2] The 2007 national blood collection and utilization survey report. Department of Health and Human Services-USA
[3] http: //www.aabb.org/Content/Donate_Blood/
[4]http://itlinkbd.com/health/blood-bank.html
[5]http://itlinkbd.com/health/blood-bank.html
[6]https://rokto.org/
[7]http://www.academia.edu/16825654/Blood_Donor_Information_and_Management_System